

THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

Résolution en vue de l'optimisation de modèles macro-économiques. Application au modèle Maribel du Bureau du Plan

Engels, John Smeets; Smeets, Monique

Award date:
1982

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES N.D. DE LA PAIX
NAMUR
FACULTE DES SCIENCES

RÉSOLUTION EN VUE DE L'OPTIMISATION
DE MODÈLES MACRO-ÉCONOMIQUES
APPLICATION AU MODÈLE
MARIBEL DU BUREAU DU PLAN

Mémoire présenté pour l'obtention du grade
de Licencié en Sciences
mathématiques
par

Monique SMEETS
John ENGELS

Année Académique 81-82

Promoteur : NGUYEN Van Hien

TOME I

Nous tenons à remercier vivement toutes les personnes qui nous ont aidés au cours de ce travail :

Monsieur Van Hien NGUYEN, qui en acceptant la direction de ce mémoire, nous a permis de réaliser un travail très enrichissant ;

Monsieur Henri BOGAERT, expert du Bureau du Plan, qui nous a fourni des explications et renseignements d'ordre économique ;

Monsieur Jean-Marc PAUL dont les suggestions nous ont été d'une grande aide ;

Monsieur Hubert CLAES pour la patience sans fin dont il a témoigné lors de la résolution de nos problèmes informatiques ;

Mademoiselle Colette SCHILS pour ses conseils apportés à la rédaction.

Nous remercions également nos parents qui nous ont permis de porter à bonne fin nos études.

"Et nunc reges, intelligite ; erudimini,
qui judicatis terram."

"Et maintenant, rois, comprenez ;
instruisez-vous, vous qui décidez du
sort de la terre."

(Paroles du Psalmiste (ps. II, 10))

TABLE DES MATIERES

TOME I

=====

INTRODUCTION	1
<u>Chapitre I</u> : GENERALITES	4
1) Qu'est-ce qu'un modèle ?	4
2) Modèles macro-économiques	5
3) Modèle MARIBEL	6
<u>Chapitre II</u> : POSITION DU PROBLEME	13
1) Simulation	13
2) Exogénéisation et endogénéisation de variables	14
3) Optimisation	15
4) Projets de recherche	16
<u>Chapitre III</u> : FORMULATION MATHEMATIQUE DES PROBLEMES	18
1) Traitement des variables, exogénéisation	18
2) Notations	30
3) Hypothèses générales et validité de l'étude	33
4) Choix de la stratégie	34
5) Etude de la structure	37
a) Notion de matrice d'incidence	37
b) Méthode CPR pour l'évaluation du jacobien	43
c) Réduction à la forme quasi-triangulaire	45
6) Méthode de l'état adjoint	49

<u>Chapitre IV</u>	: FORME QUASI-TRIANGULAIRE D'UN SYSTEME D'EQUATIONS	61
1)	Rappels de la théorie des graphes	61
2)	Détermination du prologue et de l'épilogue	65
3)	Détermination des variables de bouclage	71
a)	Caractérisation d'une variable de bouclage en termes de théorie des graphes	71
b)	Procédé théorique	73
c)	Algorithme de Ravelli	74
d)	Un autre algorithme	90
<u>Chapitre V</u>	: RESOLUTION DU MODELE	106
1)	Accès aux équations	106
2)	Sous-routine RESOLU	113
3)	Sous-routine CALFUN	114
4)	Sous-routine CALJAC	115
<u>Chapitre VI</u>	: RESOLUTION D'UNE EQUATION NON LINEAIRE A UNE INCONNUE	118
1)	Position du problème	118
2)	Quelques méthodes de résolution d'une équation non linéaire	119
3)	Algorithme utilisé	127
<u>Chapitre VII</u>	: RESOLUTION D'UN SYSTEME D'EQUATIONS NON LINEAIRES	141
1)	Méthode de Newton et ses variantes	141
2)	Méthode de Brown	144
3)	Méthode de Gauss-Seidel	146

4) La méthode des moindres carrés	147
5) Algorithme proposé par Powell	148

<u>Chapitre VIII</u> : UTILISATION DES PROGRAMMES	154
---	-----

TOME II : ANNEXES
=====

INTRODUCTION

Confrontés à la complexité croissante des problèmes de politique économique nationale, les économistes chargés d'éclairer l'évolution des agents économiques principaux ont davantage recours aux outils mathématiques sous forme de modèles macro-économiques. Ceux-ci suscitent dès lors l'intérêt tant des institutions de planification publiques et privées que des centres d'études et de prévision nationales.

Un modèle macro-économique se définit en général par un système d'équations non linéaires aux différences finies traduisant l'interdépendance et l'évolution des variables économiques principales d'un pays. On y distingue deux sortes de variables : les variables exogènes déterminées en dehors du modèle et les variables endogènes déterminées par le modèle même en fonction des exogènes. Parmi ces variables exogènes figurent entre autres les instruments de la politique économique. Ainsi le modèle permet de simuler l'impact de décisions gouvernementales particulières ; en d'autres mots, la résolution du système d'équations permet de prévoir le développement des variables endogènes en fonction d'une projection déterminée des valeurs des instruments de la politique économique. Utilisée pour prévoir le futur, la solution du modèle doit se baser sur des hypothèses d'évolution des variables exogènes.

Une deuxième catégorie de problèmes consiste à déterminer les valeurs "optimales" à donner aux instruments principalement pour résoudre un problème de politique économique posé aux manipulateurs du modèle ; ces valeurs permettraient en outre d'atteindre un objectif précis fixé par les responsables.

La résolution des deux catégories de problèmes posés ci-dessus nécessitent des algorithmes de résolution et d'optimisation performants, ainsi que des outils informatiques adéquats.

Un modèle macro-économique étant une traduction en langage mathématique (et par suite une simplification) du comportement réel de l'économie, celui-ci est soumis à des ajustements perpétuels causés par l'acquisition de nouvelles connaissances, l'apport de nouvelles données statistiques ou la prise de décisions gouvernementales. Il en résulte que le système d'équations n'est pas donné une fois pour toutes, mais évolue en même temps que l'économie. De plus, pour des problèmes particuliers le rôle de certaines variables (endogènes ou exogènes) peut être interverti, des liaisons entre variables supprimées.

Un des buts essentiels de notre travail est la mise au point d'un logiciel informatique souple permettant de générer un modèle "net" correspondant à un problème bien précis à partir du modèle "brut" contenant toutes les informations, ce logiciel tenant compte de modifications éventuelles du modèle "brut".

Un deuxième but essentiel est l'étude des méthodes proposées par Gabay et al (<2>) pour la résolution de modèles macro-économiques, ainsi que l'implémentation de celles-ci adaptées au modèle belge MARIBEL.

Actuellement, le Bureau du Plan possède déjà un algorithme efficace de résolution, mais son manque de souplesse ne permet, ni son utilisation lors d'un problème d'optimisation, ni l'inversion des rôles des variables endogènes et exogènes (problème d'exogénéisation et d'endogénéisation).

Au premier chapitre, nous introduisons le concept de modèle et de modèle macro-économique et nous donnons la description économique du modèle belge MARIBEL.

Le chapitre II sert à définir les problèmes d'un point de vue économique et à préciser les projets de recherche.

Au chapitre III, nous posons les mêmes problèmes en termes mathématiques, nous fournissons une structure informatique permettant la génération du modèle "net" à partir du modèle "brut" (y compris l'exogénéisation et l'endogénéisation de variables) et nous proposons une stratégie pour résoudre et optimiser le modèle. Celle-ci s'est inspirée des recherches

entreprises par Gabay et al. La taille du problème est réduite par la mise sous forme quasi-triangulaire du système d'équations, ce qui permet notamment d'identifier les variables de bouclage. Un algorithme de résolution d'un système non linéaire est appliqué au système réduit des équations de bouclage, et l'optimisation se fera par un algorithme utilisant le gradient réduit qui sera calculé par la méthode de l'état adjoint.

Au chapitre IV nous analysons le problème de réduction à la forme quasi-triangulaire en termes de théorie de graphes, nous exposons l'algorithme de Ravelli et nous proposons une autre méthode ayant des propriétés intéressantes pour déterminer les variables de bouclage.

Le chapitre V explique le fonctionnement des différentes sous-routines utilisées dans notre programme et suggère un procédé permettant un gain de temps calcul considérable lors de l'évaluation du jacobien du système réduit.

La mise au point d'un algorithme de résolution d'une équation non linéaire à une inconnue, nécessaire lors d'une exogénéisation fera l'objet du chapitre VI.

Au chapitre VII, nous exposons l'algorithme proposé par Powell pour la résolution d'un système d'équations non linéaires, auquel nous avons apporté quelques modifications.

En dernier lieu, au chapitre VIII, nous donnons des informations supplémentaires pour l'utilisation de nos programmes.

Chapitre I

GENERALITES

§ 0 QU'EST-CE QU'UN MODÈLE ?

Un modèle, par définition, est une représentation d'un phénomène ou d'un processus réel. Le but de la modélisation est d'expliquer, de prédire ou de contrôler le phénomène en question.

La modélisation fait partie intégrale de toute science. Ainsi la loi de la force d'attraction F entre deux corps de masse m et m' se trouvant à une distance d ,

$$F = k \frac{m m'}{d^2}$$

formulée par Newton, est une modélisation mathématique du phénomène réel de l'attraction entre deux corps. Un autre exemple de modélisation est la décomposition de la matière en molécules, atomes, noyaux, électrons, neutrons, protons...

Une modélisation est en fait une simplification de la réalité afin de la rendre manipulable. Un modèle est donc toujours un certain compromis entre la précision avec laquelle le phénomène réel est représenté et la possibilité d'utilisation qu'il donne.

De ceci résulte qu'un modèle n'est jamais un "absolu" et qu'il doit toujours être ajusté et modifié en fonction des nouvelles connaissances acquises.

Si on considère le cas particulier du modèle mathématique, il s'agit le plus souvent d'un système d'équations (linéaires, non linéaires, aux dérivées partielles,...) représentant l'état (modèle statique) ou l'évolution (modèle dynamique) d'un phénomène réel.

§ 0 MODELES MACRO-ÉCONOMIQUES

Un modèle macro-économique est en général un système d'équations non linéaires aux différences finies représentant l'interdépendance des variables économiques principales d'un pays. On distingue deux sortes de variables intervenant dans le modèle :

- les variables exogènes déterminées en dehors du modèle
par exemple : le prix du pétrole
- les variables endogènes déterminées par le modèle même
par exemple : le chômage

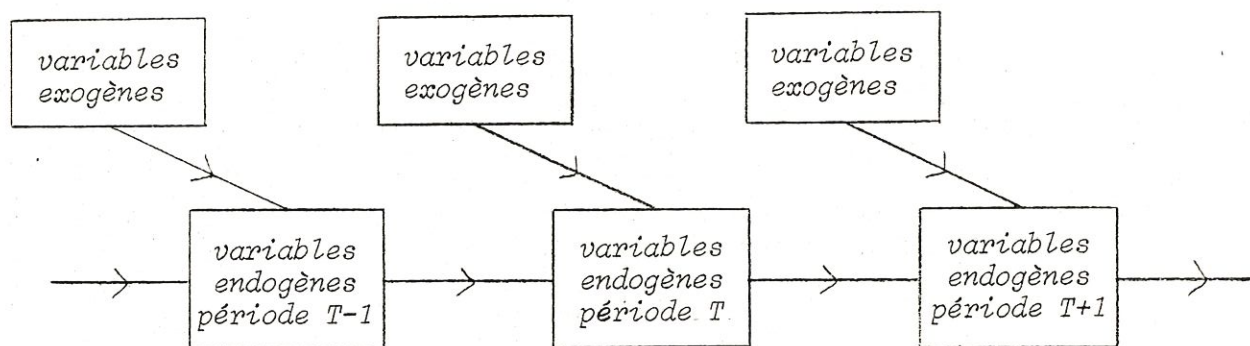
Les variables exogènes sont classées en deux groupes :

- les variables considérées comme données (en réalité, estimées ou prévues en dehors du modèle)
 - d'une part, les variables d'ajustement statistique
 - d'autre part, par exemple :
 - les variables décrivant l'environnement international pour un pays de poids international négligeable (le taux de change du franc belge en dollar,...)
 - les paramètres démographiques nationaux (la population active,...)
 - etc.
- les instruments :
ce sont les variables sur lesquelles les autorités de la politique économique peuvent agir pour atteindre des objectifs particuliers
par exemple : le taux de sécurité sociale, le taux de taxation...

Le fonctionnement d'un modèle macro-économique peut être schématisé par la figure 1.

Pour déterminer les valeurs des variables endogènes à une période T à l'aide du modèle, il suffira de connaître les valeurs des variables exogènes ainsi que celles des variables endogènes retardées (c'est-à-dire des périodes antérieures).

Remarque : Pour plus de détails, voir <19>



- figure 1 -

§ 0 LE MODÈLE MARIBEL

Le modèle MARIBEL (*"Model of Analysis and Rapid Investigation of Belgium's Economic Aggregate"*) est un modèle macro-économique dynamique, de période annuelle, non linéaire et non récursif de court/moyen terme décrivant l'économie belge. Les équations sont estimées par moindres carrés ordinaires et moindres carrés non linéaires.

Il ne possède pas de dimension régionale, ni de désagrégation sectorielle poussée : il distingue le secteur public du secteur des entreprises et isole les branches logement et gens de maison en raison de leurs comportements particuliers.

Jusqu'à présent, MARIBEL permet de simuler la formation de toute l'optique des revenus du P.N.B. ainsi que de reconstruire les comptes des opérations courantes des entreprises, des particuliers et de l'Etat et le compte des opérations en capital de l'Etat.

Le modèle comporte environ 450 variables endogènes et une soixantaine de variables exogènes.

En annexe, on trouvera :

- une liste de toutes les variables intervenant dans le modèle
- les équations se trouvant dans la sous-routine MODEL (écrite en Fortran)

Les principaux mécanismes du modèle sont les suivants :

a. Capacité de production et facteurs de production

Le secteur des entreprises est caractérisé par une fonction de production à deux facteurs, capital et travail : il s'agit d'une fonction qui lie la productivité à l'intensité capitalistique et à un terme de progrès autonome. Un mécanisme d'ajustement partiel permet d'identifier des élasticités de long terme et de court terme de la production aux facteurs.

En introduisant les facteurs de production potentiels (population active au lieu du nombre d'emplois, notamment), on obtient à l'aide de cette fonction la capacité de production des entreprises qui, à long terme, est principalement liée à l'évolution du stock de capital.

D'un autre côté, la production effective est déterminée par la demande qui s'adresse aux entreprises et correspond à la valeur ajoutée de ce secteur.

Le taux d'utilisation des capacités de production qui est obtenu à partir de ces deux notions est une variable importante de tension conjoncturelle qui joue à de nombreuses reprises dans le modèle, principalement sur les prix.

L'évolution des facteurs de production découle essentiellement de la demande à laquelle doivent répondre les entreprises.

L'investissement y est directement lié par le mécanisme de l'accélérateur, mais il dépend aussi à court terme d'un indicateur des bénéfices des entreprises qui reflète à la fois l'impact des possibilités d'autofinancement et les attentes en matière de profit.

La demande de travail s'ajuste au niveau de la demande globale moyennant les contraintes techniques de la fonction de production. Elle se répartit entre un nombre de travailleurs et une durée du travail presté en fonction de l'évolution des salaires (par rapport au prix de la valeur ajoutée) et des pressions qui existent sur le marché de l'emploi.

Le coût du facteur capital dépend du prix des investissements, des taux d'intérêt, et des subventions accordées par le secteur public. Le taux de salaire réel est fondamentalement lié à la productivité, mais il dépend aussi de la situation du marché de l'emploi.

b. Les éléments de la demande finale et le commerce extérieur

Les prix de principaux éléments de la demande finale (consommation privée, investissement productifs ou en logement ou publics) sont déterminés pour l'essentiel par leurs composantes coûts, mais sont aussi influencés à court terme par les tensions conjoncturelles comme cela a été dit ci-dessus.

La consommation privée est liée au revenu disponible par une fonction basée sur l'hypothèse du revenu permanent de FRIEDMAN.

Les investissements en logement dépendent non seulement du revenu disponible, mais aussi des taux d'intérêt réels.

La demande du secteur public est formée, d'une part, par la consommation publique et, d'autre part, par les investissements publics. Le volume de la consommation publique est directement liée à l'évolution du nombre de personnes occupées dans le secteur public. Les investissements publics sont exogènes en valeur et leur volume dépend donc de l'évolution des prix.

Le volume des exportations de biens est déterminé par l'évolution quantitative du commerce international, par la position de la Belgique sur les marchés internationaux, et par le rapport entre les prix belges à l'exportation et les prix mondiaux, compte tenu de l'évolution du taux de change du F.B. Le modèle utilisé suppose qu'à long terme le rythme de croissance des exportations belges et celui du volume du commerce mondial convergent. Les prix des exportations s'ajustent à long terme sur les prix mondiaux, mais à court terme ils sont influencés par les hausses des coûts intérieurs.

Les importations sont liées à la somme des composantes de la demande finale pondérée par leur contenu en importations et elles dépendent aussi du rapport entre les prix des biens intérieurs et le prix des biens importés. Celui-ci est calculé par un ajustement sur les prix pratiqués sur les marchés internationaux.

Les exportations et importations de services, ainsi que leurs prix, ont été liées à l'évolution des échanges des biens, et de leurs prix.

La seule exception concerne les importations de tourisme qui sont traitées de manière analogue à la consommation privée.

c. Les revenus et les comptes d'agents

La rémunération des facteurs de production donne les principaux revenus primaires des divers agents (salariés, indépendants, entreprises). A cela s'ajoute les revenus de propriété et d'entreprises (dont les intérêts de la dette publique).

Les prélèvements fiscaux et parafiscaux sont effectués au profit des différents secteurs de l'Etat qui, à leur tour, font des transferts de revenus vers les autres agents.

Une fois que toutes les opérations de redistribution sont effectuées, on peut, à l'aide du modèle, reconstituer entièrement l'optique revenus du P.N.B., et les comptes des opérations courantes de tous les agents (y compris celui du secteur des entreprises).

Une attention toute particulière a été apportée au traitement des comptes de l'Etat dont les différentes composantes sont analysées avec un maximum de détails que ce soit au niveau du pouvoir central, des pouvoirs subordonnés et de la sécurité sociale, où chaque régime est traité isolément.

d. Les aspects financiers et monétaires du modèle

Le modèle n'a pas été conçu pour prendre en compte les flux financiers et monétaires entre les divers agents. Il contient néanmoins un module monétaire qui permet de lier la masse monétaire aux besoins de financement du secteur privé et du secteur public, et aussi à l'état de la balance des paiements, et qui détermine les taux d'intérêt à partir du coefficient de liquidité de l'économie.

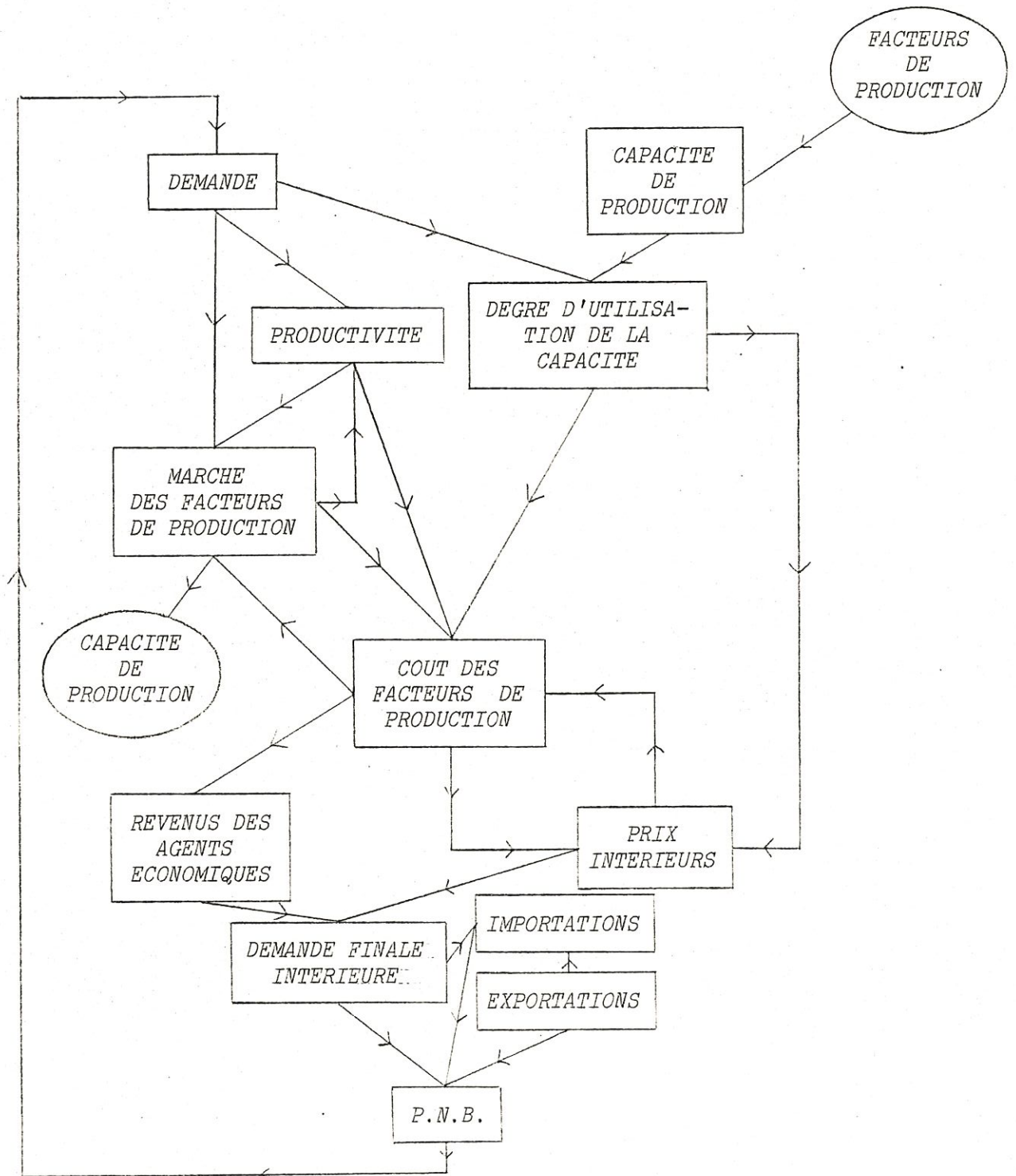
On notera encore que la construction d'un compte complet des opérations en capital de l'Etat (à partir du regroupement économique des dépenses du pouvoir central notamment) contribue grandement à une évaluation plus

correcte du besoin de financement de l'Etat, de la dette publique et de son impact sur le reste de l'économie.

Tiré de <16>

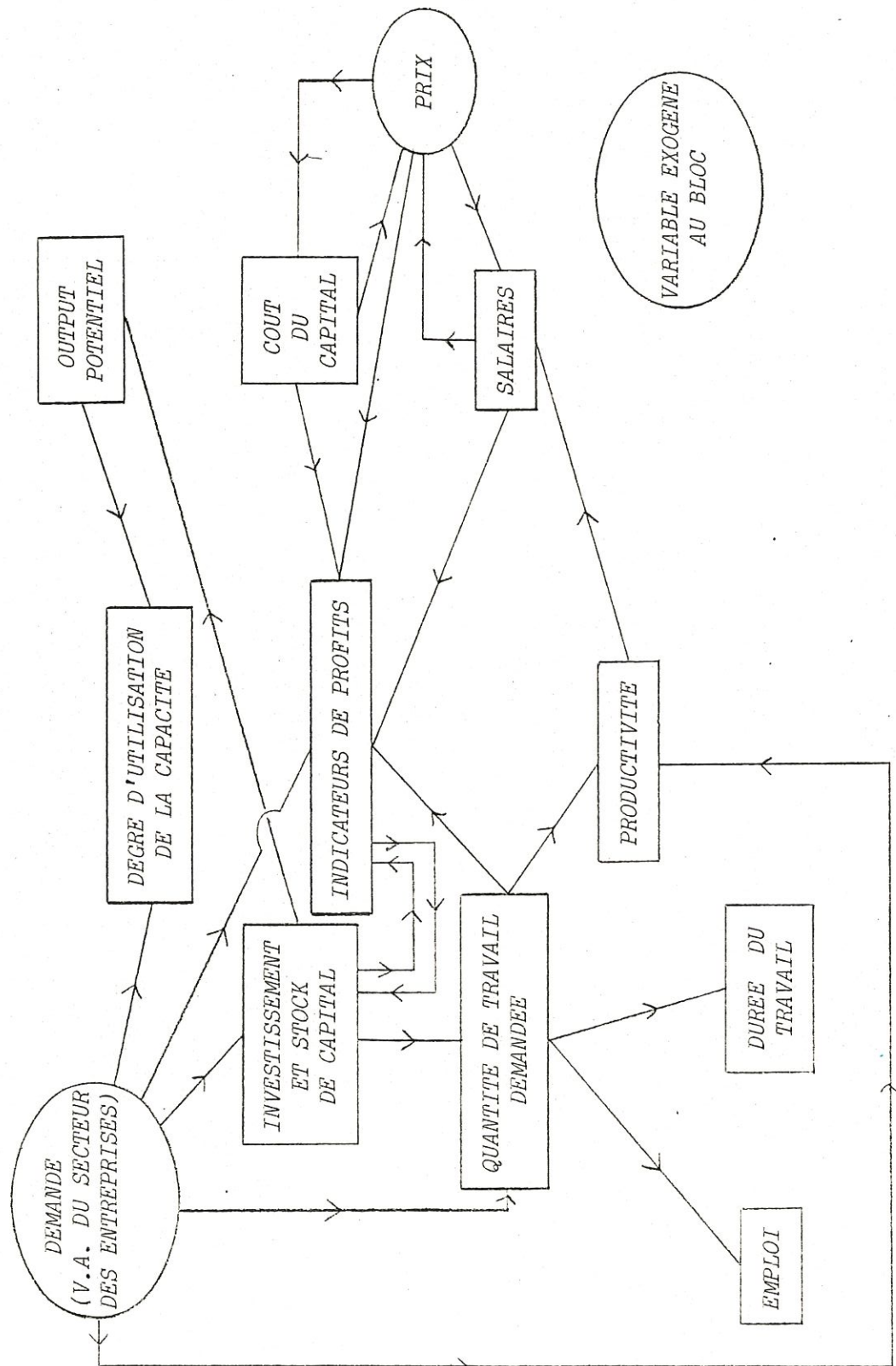
Les diagrammes qui suivent présentent schématiquement les principales relations du modèle et de son bloc central (figure 2 et figure 3).

LES GRANDS CIRCUITS D'INTERDEPENDANCE DE MARIBEL



- figure 2 -

LES RELATIONS DU BLOC PRODUCTION, FACTEURS DE PRODUCTION, COUTS (SECTEURS DES ENTREPRISES)



- figure 3 -

Chapitre II

POSITION DU PROBLEME

§ 0 SIMULATION

Nous avons vu dans le chapitre I qu'un modèle permet de déterminer les valeurs des variables endogènes dès que les valeurs des variables exogènes sont fixées. Pour ceci, il suffit de résoudre le système d'équations à la période donnée. Or, parmi les variables exogènes figurent les instruments de la politique économique. Pour les autorités il sera alors intéressant de connaître l'impact de leurs décisions sur l'évolution de l'économie belge, c'est-à-dire de donner une réponse à la question suivante : "*Que se passera-t-il, si ...*" (par exemple, si on augmente le taux de sécurité sociale..., ou si on diminue les taux de taxation). Le modèle MARIBEL permet de simuler l'évolution de l'économie dans ces conditions (il ne faudra donc pas attendre trois ans, pour savoir ce qui ne se passera pas en 1985...). En clair, *une simulation consiste donc à donner des valeurs particulières aux instruments (exogènes) et de déterminer à l'aide du modèle les valeurs correspondantes des variables endogènes*, ceci pouvant se faire pour n'importe quel nombre d'années, par exemple de l'année 1982 à l'année 1986. Il suffira de résoudre les équations du modèle successivement de l'année 1982 à l'année 1986, les valeurs obtenues pour une année servant de base à la détermination des valeurs pour les années ultérieures.¹

Pour faire une simulation, il faut donc disposer d'un algorithme de résolution d'un système d'équations non linéaires.

¹ Comme nous avons affaire à un modèle probabiliste, plus la prévision est éloignée dans le temps, plus l'écart-type des variables est élevé.

§ 0 EXOGENÉISATION ET ENDOGENÉISATION DE VARIABLES

On ne peut pas toujours déterminer a priori quelles variables seront endogènes et lesquelles seront exogènes. Imaginons le cas d'une équation liant deux variables dont une est nécessairement exogène et l'autre endogène. On peut fixer celle qui sera endogène et celle qui sera exogène.

Dans la pratique, il peut aussi arriver qu'on doive exogénéiser une variable antérieurement endogène. Prenons le cas des dernières décisions gouvernementales : le blocage des prix. Avant de prendre cette décision, il fallait se rendre compte de l'effet d'un blocage des prix. Si le modèle avait permis d'exogénéiser la variable "prix", normalement endogène, il aurait été possible de faire une simulation de cette mesure.

Quand on exogénéise une variable endogène, il faut nécessairement supprimer une équation ou endogénéiser une variable exogène, car sinon le système est surdéterminé. En pratique, dans le cas où on supprime une équation, ce n'est pas nécessairement celle qui déterminait la variable maintenant exogénéisée.

Un autre cas d'exogénéisation se présente fréquemment. Un tel modèle a besoin d'une énorme banque de données contenant entre autres les valeurs des variables exogènes et endogènes des années écoulées. Cette banque de données est mise à jour au fur et à mesure que les bureaux de statistique fournissent leurs résultats. Il arrive alors à un moment donné qu'on connaisse déjà les valeurs réelles de certaines variables endogènes sans les connaître pour d'autres pour la même année. Pour tenir compte de ces valeurs réelles, il faudra donc exogénéiser ces variables pour la première année de simulation.

Comme les problèmes d'exogénéisation et d'endogénéisation sont fort importants, il faudra donc que l'algorithme de résolution puisse en tenir compte.

§ 9 OPTIMISATION

Nous avons déjà vu qu'une simulation permet de répondre à la question : "Que se passera-t-il si on modifie la valeur d'un instrument ?" Une autre question que les responsables de la politique pourraient se poser, est la suivante : *"Quelles valeurs optimales faut-il donner aux instruments pour atteindre un certain objectif ?"*, par exemple pour réduire le chômage, réduire le déficit budgétaire, équilibrer la balance des paiements...

Ici, contrairement à la simulation, les valeurs des instruments ne sont pas fixées dès le départ ; au contraire, le problème est justement de trouver leurs valeurs en fonction d'un certain critère d'optimalité.

Nous sommes donc en présence d'un problème d'optimisation, dont la fonction objectif sera un but économique précis, les variables de contrôle seront les instruments et les contraintes seront les liens entre les agents économiques (c'est-à-dire les équations du modèle).

On distingue trois classes de problèmes d'optimisation :

- minimiser l'écart d'une ou de plusieurs variables à une trajectoire fixée
par exemple, diminuer le chômage de 10 % chaque année
- minimiser l'écart d'une ou de plusieurs variables à des valeurs fixées pour une période T donnée, sans imposer de trajectoire
par exemple, diminuer le chômage de 50 % jusqu'en 1987, sans pour autant imposer une diminution de 10 % chaque année
- minimiser le temps pour réduire l'écart d'une ou de plusieurs variables à des valeurs fixées
par exemple, minimiser le temps pour réduire le chômage de 50 %

§ 0 PROJETS DE RECHERCHE

Pour répondre à tous ces problèmes, il faut donc avoir les outils suivants à sa disposition :

- une banque de données regroupant toutes les informations nécessaires
- des programmes de mise à jour et de manipulation de ces données
- un algorithme efficace de résolution d'un système d'équations non linéaires de grande taille
- une structure mathématique permettant facilement l'exogénéisation et l'endogénéisation totale (c'est-à-dire pour toutes les années de simulation) de certaines variables
- une structure mathématique permettant l'exogénéisation et l'endogénéisation partielle (c'est-à-dire pour la première année de simulation) de certaines variables
- un algorithme efficace d'optimisation
- un software facilement utilisable regroupant toutes ces possibilités

Actuellement, le Bureau du Plan possède déjà tout ce qui concerne les données et leur traitement, ainsi qu'un programme permettant de faire des simulations "simples".

Jusqu'à présent l'exogénéisation partielle se fait en introduisant manuellement des tests supplémentaires dans les équations même du modèle, tandis que l'exogénéisation ou l'endogénéisation totale est encore impossible.

Le but de ce travail consiste d'abord à développer une structure informatique permettant facilement l'exogénéisation et l'endogénéisation totale de variables.

Puisque chaque itération de l'optimisation nécessite une simulation complète, il est important d'avoir un algorithme de résolution très efficace ; il sera d'autant plus efficace que l'on pourra tirer profit de résultats intermédiaires de l'algorithme de résolution pour épargner des calculs lors de l'optimisation. Nous allons proposer un algorithme de résolution qui s'intégrera de manière très efficace dans un programme d'optimisation.

L'implémentation sur ordinateur des algorithmes proposés fait entièrement partie du travail.

Nous développerons aussi un modèle d'optimisation sans pour autant donner d'algorithme particulier.

Vu le temps limité, ceci dépasserait le cadre de ce travail.

Chapitre III

FORMULATION MATHEMATIQUE DES PROBLEMES

§ 0 TRAITEMENT DES VARIABLES, EXOGÉNÉISATION

Soit un modèle macro-économique liant N variables par $Nequa$ équations.

Remarque : Dans ce travail, nous n'envisagerons pas comment on construit un modèle macro-économique, ceci relevant du travail d'économètres. (voir <19>)

Si pour des raisons de simplification nous ne tenons pas compte des variables retardées, le système d'équations pour une période donnée peut s'écrire :

$$g_i(x_1, x_2, \dots, x_N) = 0 \quad i=1, 2, \dots, Nequa$$

où x_j sont les variables économiques (endogènes ou exogènes).

La plupart des équations d'un système macro-économique étant des équations de définition ou des équations de comportement, c'est-à-dire expliquant le comportement d'une variable en fonction d'autres, ces équations ont la forme explicite :

$$y_i = f_i(x_1, x_2, \dots, x_N)$$

où y_i est généralement la $i^{\text{ième}}$ variable endogène (y_i est donc un des x_j).

En pratique, à l'intérieur d'un programme de résolution, il faudra pouvoir évaluer les équations. En Fortran, il faudra alors nécessairement faire une assignation. Le cas où l'équation n'a pas la forme explicite

ci-dessus, n'est donc pas traitable comme tel. On transformera l'équation en question de la manière suivante :

$$y_i = y_i + g_i(x_1, x_2, \dots, x_N)$$

Pour le modèle MARIBEL, les équations se trouvent dans une sous-routine appelée MODEL.

Chaque équation a donc sa variable d'assignation (en général, la variable endogène correspondant à l'équation de comportement ou de définition).

Le programme du Bureau du Plan range les variables par ordre alphabétique à l'intérieur de deux classes de variables : exogènes et endogènes. Comme pour le cas de MARIBEL toutes les équations sont des équations de comportement ou de définition, il s'en suit que la variable d'assignation de la $i^{\text{ième}}$ équation est la $i^{\text{ième}}$ variable endogène, ce qui est fort commode. D'autre part, ce système exige le classement a priori des variables endogènes et exogènes et rend une exogénéisation impossible.

Pour ces raisons, nous rangeons les variables par ordre alphabétique sans faire de distinction a priori entre variables endogènes et variables exogènes. Dans notre programme, les valeurs de toutes les variables sont stockées dans une matrice Y (R dans MODEL), une colonne contenant les valeurs d'une même variable pour les différentes périodes. Il s'en suit, que la variable d'assignation de la $i^{\text{ième}}$ équation n'est plus nécessairement la $i^{\text{ième}}$ variable. Il faudra donc ajouter un vecteur, soit IWATEQ, contenant cette information :

$$IWATEQ(I) = J$$

(1)

ssi

J est le numéro de la variable d'assignation de l'équation I

Illustrons le vecteur IWATEQ sur le modèle macro-économique TOTO.

TOTO fait intervenir 7 variables : a, b, c, d, e, o, p

dont 4 endogènes de départ : a, b, d, p

et 3 exogènes de départ : c, e, o

Voici les équations du modèle :

$$\left\{ \begin{array}{l} a = g_1(c, e) \\ b = g_2(a, d, e, o, p) \\ d = g_3(a, b, e, o) \\ p = g_4(c, d) \end{array} \right.$$

Le programme du Bureau du Plan range les variables comme suit :

- endogènes : $x_1 = a$

$$x_2 = b$$

$$x_3 = d$$

$$x_4 = p$$

- exogènes : $z_1 = c$

$$z_2 = e$$

$$z_3 = o$$

Le modèle peut donc se réécrire de la manière suivante :

$$\left\{ \begin{array}{l} x_1 = g_1(z_1, z_2) \\ x_2 = g_2(x_1, x_3, x_4, z_2, z_3) \\ x_3 = g_3(x_1, x_2, z_2, z_3) \\ x_4 = g_4(x_3, z_1) \end{array} \right.$$

Nous constatons que les numéros des variables d'assignation sont les numéros des équations correspondantes.

Dans notre programme les variables sont rangées de la manière suivante :

$$y_1 = a$$

$$y_2 = b$$

$$y_3 = c$$

$$y_4 = d$$

$$y_5 = e$$

$$y_6 = o$$

$$y_7 = p$$

ce qui permet de mettre le système sous la forme

$$\begin{cases} y_1 = g_1(y_3, y_5) \\ y_2 = g_2(y_1, y_4, y_5, y_6, y_7) \\ y_4 = g_3(y_1, y_2, y_5, y_6) \\ y_7 = g_4(y_3, y_4) \end{cases}$$

dans lequel y_1, y_2, y_4, y_7 sont endogènes
et y_3, y_5, y_6 sont exogènes.

Le vecteur IWATEQ défini en (1), prendra les valeurs :

$$IWATEQ(1) = 1$$

$$IWATEQ(2) = 2$$

$$IWATEQ(3) = 4$$

$$IWATEQ(4) = 7$$

$IWATEQ(4) = 7$ car y_7 est la variable d'assignation de l'équation 4.

Remarque : Lors de la construction des équations du modèle, les variables d'assignation étaient en quelque sorte des variables "endogènes a priori" (car l'équation exprime la variation de la variable d'assignation en fonction d'autres variables). Cependant, il arrivera couramment pour un problème particulier qu'une variable d'assignation soit exogène (exogénéisation).

Il est évident qu'il faut avoir autant de variables endogènes que d'équations utilisées (ce qui était automatiquement réalisé à la construction). Il est donc possible d'établir une correspondance biunivoque entre les variables endogènes et les équations utilisées. Si la variable d'assignation d'une équation est endogène, cette variable correspond de façon naturelle à l'équation en question.

On verra dans la suite qu'on peut tirer profit de cette correspondance biunivoque. Il sera donc utile de l'établir même si elle ne s'impose pas de façon naturelle, c'est-à-dire pour le problème d'une exogénéisation.

Pour fixer les idées, supposons qu'on exogénéise la variable d'assignation de l'équation I.

Trois cas peuvent maintenant se présenter :

- 1) On supprime l'équation I.

Il ne serait pas profitable de devoir modifier chaque fois la sous-routine MODEL. Nous verrons dans la suite qu'on peut s'en passer.

- 2) On endogénéise une variable antérieurement exogène, soit J le numéro de cette variable.

Dans ce cas, nous faisons correspondre la variable J à l'équation I.

Deux possibilités :

- a) La variable J ne figure pas dans l'équation I.

Nous reviendrons sur cette possibilité plus tard

- b) La variable J y figure.

Dans ces conditions, il sera avantageux de pouvoir tirer la valeur de la variable J de son équation (I) sans pour autant devoir transformer la sous-routine MODEL afin d'avoir la variable J comme variable d'assignation.

Par conséquent, il sera nécessaire de mettre au point un algorithme efficace de résolution d'une équation non linéaire par rapport à une inconnue. Ceci fera l'objet du chapitre VI.

- 3) On supprime une autre équation, soit K.

Supposons que la variable endogène correspondante soit la variable L.

Nous avons alors une équation sans variable (équation I) et une variable endogène sans équation (la variable L).

Dès lors, nous allons faire correspondre la variable endogène L à l'équation I.

Ici on retrouve les deux possibilités a) et b) explicitées plus haut.

Pour le problème d'optimisation, il faut préciser les variables de contrôle, c'est-à-dire les instruments économiques sur lesquels les responsables aimeraient agir pour atteindre leur objectif. Nous appelons donc *contrôle* une variable exogène dont la valeur est à déterminer lors d'un problème d'optimisation précis. Les autres variables exogènes seront considérées comme données. Dans le cas d'une simulation, toutes les variables exogènes sont donc des données.

Toutes ces informations seront contenues dans un vecteur d'identification des N variables, soit $IDVAR$:

$$(2) \quad IDVAR(J) = \begin{cases} 2000 & \text{si la variable } J \text{ est une donnée} \\ & \text{(soit } N_{don} \text{ le nombre de ces données)} \\ 1000 & \text{si la variable } J \text{ est un contrôle} \\ & \text{(soit } N_{ins} \text{ le nombre de contrôles)} \\ I & \text{si la variable } J \text{ correspond à l'équation } I \\ & \text{et qu'elle y figure} \\ -I & \text{si la variable } J \text{ correspond à l'équation } I \\ & \text{sans y figurer} \end{cases}$$

Appelons N_{endo} le nombre de variables endogènes.

Remarque : Si une équation I est supprimée, il n'y a pas de variable correspondante. $IDVAR$ ne prendra donc jamais ni la valeur I , ni la valeur $-I$.

Cette information est donc bien donnée par le vecteur $IDVAR$.

Reprenons l'exemple du modèle TOTO pour illustrer les différents cas pouvant se présenter lors d'une exogénéisation et la construction du vecteur $IDVAR$ correspondant à chacun de ces cas.

I Cas initial : y_1, y_2, y_4, y_7 sont endogènes

y_3, y_5, y_6 sont exogènes et des données

Les composantes du vecteur $IDVAR$ correspondant aux variables endogènes valent le numéro de l'équation correspondante, tandis que celles correspondant aux variables exogènes valent 2000.

$IDVAR$ vaut dans ce cas :

$$IDVAR(1) = 1$$

$$IDVAR(2) = 2$$

$$IDVAR(3) = 2000$$

$$IDVAR(4) = 3$$

$$IDVAR(5) = 2000$$

$$IDVAR(6) = 2000$$

$$IDVAR(7) = 4$$

II On veut exogénéiser la variable d'assignation de l'équation 3 en tant que donnée, c'est-à-dire y_4

Comme y_4 est devenu exogène, IDVAR(4) vaut maintenant 2000.

Illustrons les trois cas envisagés plus haut :

1) On supprime l'équation 3.

Le vecteur IDVAR prend alors les valeurs suivantes :

$$\text{IDVAR}(1) = 1$$

$$\text{IDVAR}(2) = 2$$

$$\text{IDVAR}(3) = 2000$$

$$\text{IDVAR}(4) = 2000$$

$$\text{IDVAR}(5) = 2000$$

$$\text{IDVAR}(6) = 2000$$

$$\text{IDVAR}(7) = 4$$

Le vecteur IDVAR ne prenant jamais la valeur 3, on voit que c'est l'équation 3 qui a été supprimée.

2) On endogénéise une variable antérieurement exogène et on la fait correspondre à l'équation 3.

a) On endogénéise y_3 , mais elle ne figure pas dans l'équation 3.

IDVAR(3) prendra donc la valeur -3.

On ne supprime pas d'équations.

Ici, IDVAR s'écrit :

$$\text{IDVAR}(1) = 1$$

$$\text{IDVAR}(2) = 2$$

$$\text{IDVAR}(3) = -3$$

$$\text{IDVAR}(4) = 2000$$

$$\text{IDVAR}(5) = 2000$$

$$\text{IDVAR}(6) = 2000$$

$$\text{IDVAR}(7) = 4$$

b) On endogénéise y_6 qui elle figure dans l'équation 3, sans supprimer une équation.

Comme l'équation 3 correspond à la variable 6 et que celle-ci y figure, on aura donc IDVAR(6) = 3.

$$IDVAR(1) = 1$$

$$IDVAR(2) = 2$$

$$IDVAR(3) = 2000$$

$$IDVAR(4) = 2000$$

$$IDVAR(5) = 2000$$

$$IDVAR(6) = 3$$

$$IDVAR(7) = 4$$

3) On supprime une autre équation.

a) On supprime l'équation 1.

La variable correspondant à l'équation 1 est y_1 .

La variable y_1 est sans équation et l'équation 3 est sans variable (car y_4 est devenu exogène).

On fait donc correspondre la variable y_1 à l'équation 3.

Comme elle y figure, nous aurons $IDVAR(1) = 3$.

$IDVAR$ s'écrit donc :

$$IDVAR(1) = 3$$

$$IDVAR(2) = 2$$

$$IDVAR(3) = 2000$$

$$IDVAR(4) = 2000$$

$$IDVAR(5) = 2000$$

$$IDVAR(6) = 2000$$

$$IDVAR(7) = 7$$

$IDVAR$ montre bien que l'équation 1 est supprimée, car il ne vaut jamais 1.

b) On supprime l'équation 4.

On fait donc correspondre y_7 à l'équation 3. Mais y_7 n'y figure pas. Il s'en suit que $IDVAR(7)$ vaut -3.

Le vecteur $IDVAR$ prend donc les valeurs suivantes :

$$IDVAR(1) = 1$$

$$IDVAR(2) = 2$$

$$IDVAR(3) = 2000$$

$$IDVAR(4) = 2000$$

$$IDVAR(5) = 2000$$

$$IDVAR(6) = 2000$$

$$IDVAR(7) = -3$$

Remarquons que IDVAR ne prend pas la valeur 4.

III Jusqu'à présent les variables exogènes étaient toujours des données.
Nous allons examiner deux cas qui font intervenir des variables de contrôle.

1) Prenons le cas initial (I) où y_3 et y_5 sont maintenant des contrôles, c'est-à-dire

variables endogènes : y_1, y_2, y_4, y_7

variables exogènes : contrôles : y_3 et y_5

donnée : y_6

Les composantes 3 et 5 de IDVAR prendront alors la valeur 1000.

Dans ce cas, IDVAR s'écrit :

$$\text{IDVAR}(1) = 1$$

$$\text{IDVAR}(2) = 2$$

$$\text{IDVAR}(3) = 1000$$

$$\text{IDVAR}(4) = 3$$

$$\text{IDVAR}(5) = 1000$$

$$\text{IDVAR}(6) = 2000$$

$$\text{IDVAR}(7) = 4$$

2) Soit le cas initial où y_5 est contrôle.

Imaginons qu'on veuille exogénéiser y_4 en tant que contrôle sans supprimer l'équation 3.

Supposons qu'on supprime une autre équation, soit l'équation 2.

On fait alors correspondre l'équation 3 à la variable y_2 qui y figure.

IDVAR s'écrit maintenant :

$$\text{IDVAR}(1) = 1$$

$$\text{IDVAR}(2) = 3$$

$$\text{IDVAR}(3) = 2000$$

$$\text{IDVAR}(4) = 1000$$

$$\text{IDVAR}(5) = 1000$$

$$\text{IDVAR}(6) = 2000$$

$$\text{IDVAR}(7) = 4$$

Remarquons que le vecteur *IWATEQ* reste invariant tant qu'on ne change pas les équations du modèle, tandis que l'utilisateur doit fournir le vecteur *IDVAR* qui dépend du problème concret à résoudre.

Suite à l'existence de la correspondance biunivoque entre variables endogènes et équations, nous pouvons réécrire le système macro-économique sous la forme :

$$(3) \quad k_i y_i = f_i(x_1, x_2, \dots, x_N) \quad i=1, 2, \dots, N_{endo}$$

avec

$$k_i = \begin{cases} 1 & \text{si la variable endogène } y_i \text{ correspondant à} \\ & \text{l'équation } i \text{ y figure} \\ 0 & \text{sinon} \end{cases}$$

Ceci est une notation mathématique introduite pour faciliter les écritures et les développements ultérieurs.

Il est clair que dans la sous-routine *MODEL* les équations n'ont pas nécessairement cette forme-ci, comme nous l'avons d'ailleurs mentionné plus haut.

Reprenons encore l'exemple du modèle *TOTO* et envisageons le cas II 3)b) où

$$\begin{aligned} IDVAR(1) &= 1 \\ IDVAR(2) &= 2 \\ IDVAR(3) &= 2000 \\ IDVAR(4) &= 2000 \\ IDVAR(5) &= 2000 \\ IDVAR(6) &= 2000 \\ IDVAR(7) &= -3 \end{aligned}$$

Ici, *Nendo* vaut 3, c'est-à-dire que nous avons 3 équations et 3 variables endogènes.

Les équations peuvent s'écrire :

$$(4) \quad \begin{cases} y_1 = g_1(y_3, y_5) \\ y_2 = g_2(y_1, y_4, y_5, y_6, y_7) \\ y_4 = g_3(y_1, y_2, y_5, y_6) \end{cases}$$

où y_1 , y_2 et y_7 sont endogènes

y_3 , y_4 , y_5 , y_6 sont des données.

Le système (4) n'est pas de la forme (3) car la variable endogène y_7 ne figure pas dans le membre de gauche.

Cependant la 3^{ème} équation peut se réécrire de la manière suivante :

$$k_3 y_7 = f_3(y_1, y_2, y_5, y_6)$$

où $k_3 = 0$

$$f_3 = g_3 - y_4$$

Nous n'allons évidemment pas changer la sous-routine MODEL pour chaque problème particulier.

Pour évaluer la valeur de la fonction f_3 en un point, nous stockons la valeur de y_4 dans une variable intermédiaire yint, nous évaluons la fonction $y_4 = g_3$, nous calculons f_3 en faisant la différence entre y_4 et yint et nous remettons ensuite la valeur de yint dans y_4 .

Envisageons maintenant le cas II 3)a)

où

$$IDVAR(1) = 3$$

$$IDVAR(2) = 2$$

$$IDVAR(3) = 2000$$

$$IDVAR(4) = 2000$$

$$IDVAR(5) = 2000$$

$$IDVAR(6) = 2000$$

$$IDVAR(7) = 4$$

Ici aussi Nendo vaut 3.

Les équations s'écrivent :

$$(5) \quad \begin{cases} y_2 = g_2(y_1, y_4, y_5, y_6, y_7) \\ y_4 = g_3(y_1, y_2, y_5, y_6) \\ y_7 = g_4(y_3, y_4) \end{cases}$$

où y_1, y_2 et y_7 sont endogènes

y_3, y_4, y_5, y_6 sont des données.

Le système (5) n'est pas non plus de la forme (3).

Mathématiquement le système peut se réécrire :

$$\begin{cases} k_1 y_2 = f_1(y_1, y_4, y_5, y_6, y_7) \\ k_2 y_1 = f_2(y_2, y_4, y_5, y_6) \\ k_3 y_7 = f_3(y_3, y_4) \end{cases}$$

où

$$k_1 = k_2 = k_3 = 1$$

$$f_1 = g_2$$

$$f_3 = g_4$$

f_2 n'est pas connu explicitement

En effet, f_2 est la fonction $R^4 \rightarrow R$
 $(y_2, y_4, y_5, y_6) \rightarrow y_1$

tel que

$$h(y_1, y_2, y_4, y_5, y_6) = 0$$

$$\text{où } h(y_1, y_2, y_4, y_5, y_6) = g_3(y_1, y_2, y_5, y_6) - y_4$$

Pour évaluer la valeur de la fonction f_2 en un point (c'est-à-dire y_2, y_4, y_5, y_6 fixés), il faudra donc annuler la fonction h qui ne dépend plus que de y_1 .

Un algorithme de résolution d'une équation non linéaire par rapport à une variable sera proposé au chapitre VI.

Nous avons déjà fait remarquer que pour chaque problème particulier à résoudre l'utilisateur doit fournir le vecteur *IDVAR*. (Nous supposons que le vecteur *IWATEQ* est donné une fois pour toutes.)

En fait, le vecteur *IDVAR* spécifie entièrement le problème posé : il précise quelles variables sont endogènes, contrôles et données et indique la correspondance entre variables endogènes et équations.

§ 0 NOTATIONS

- a) Notons y^i la variable endogène correspondant à l'équation i et u^j le $j^{\text{ième}}$ contrôle.

En considérant les données comme constantes intervenant dans les équations, le système (3) peut se réécrire comme suit :

$$(6) \quad k_i y^i = f^i(y^1, \dots, y^{Nendo}, u^1, \dots, u^{Nins}) \quad i=1, 2, \dots, Nendo$$

Réintroduisons le caractère dynamique du modèle en tenant compte des variables retardées (négligées au paragraphe précédent).

Considérons donc un modèle dynamique en temps discret pour des périodes t allant de 1 à T .

Notons y_t^i la valeur de la variable endogène i à la période t ,

$y_t \in R^{Nendo}$ le vecteur $(y_t^1, y_t^2, \dots, y_t^{Nendo})$ et

$y \in R^{Nt}$ le vecteur (y_1, y_2, \dots, y_T) , avec $Nt = Nendo \cdot T$.

Notons u_t^j la valeur du contrôle j à la période t ,

$u_t \in R^{Nins}$ le vecteur $(u_t^1, u_t^2, \dots, u_t^{Nins})$ et

$u \in R^{Ct}$ le vecteur (u_1, u_2, \dots, u_T) , avec $Ct = Nins \cdot T$.

Nous appellerons p et q les retards maxima apparaissant dans le modèle respectivement sur les variables endogènes et sur les variables de contrôle.

Rappelons (voir figure 1) que, pour chaque période t , le modèle est donné par $Nendo$ équations, chacune dépendant des variables endogènes $y_t, y_{t-1}, \dots, y_{t-p}$, des variables de contrôle $u_t, u_{t-1}, \dots, u_{t-q}$ et des autres variables exogènes considérées comme données (dans la suite, nous ne parlerons plus de ces dernières).

Dorénavant, nous écrirons le modèle sous sa forme définitive :

$$(7) \quad \begin{aligned} k_i y_t^i &= f_t^i(y_t, y_{t-1}, \dots, y_{t-p}, u_t, u_{t-1}, \dots, u_{t-q}) \\ \text{pour } i &= 1, 2, \dots, Nendo \\ t &= 1, 2, \dots, T \end{aligned}$$

ou encore :

$$(8) \quad \begin{aligned} K y_t &= f_t(y_t, y_{t-1}, \dots, y_{t-p}, u_t, u_{t-1}, \dots, u_{t-q}) \\ \text{pour } t &= 1, 2, \dots, T \end{aligned}$$

où nous avons noté K la matrice diagonale de dimension $Nendo \times Nendo$ dont le $i^{i\text{ème}}$ élément diagonal est k_i i.e.

$$(9) \quad K = \begin{pmatrix} k_1 & & & & \\ & k_2 & & & \\ & & \ddots & & \\ & & & k_i & \\ & & & & \ddots \\ & & & & & k_{Nendo} \end{pmatrix}$$

avec y_t donné pour $t = -1, -2, \dots, 1-p$
 u_t donné pour $t = -1, -2, \dots, 1-q$
 k_i valant 0 ou 1

Remarque : Pour faciliter les écritures, dans la suite nous omettrons parfois l'indice t de la période.

Dans ce cas, nous notons y_i (au lieu de y_t^i) la $i^{\text{ième}}$ variable endogène.

- b) Nous supposons que le critère, ou la fonction objectif, soit donné sous la forme générale suivante :

$$(10) \quad j(y,u) = \sum_{t=1}^T j_t(y_t, y_{t-1}, \dots, y_{t-p}, u_t, u_{t-1}, \dots, u_{t-q})$$

où les fonctions j_t , fonctions coût instantané, sont données par l'utilisateur.

Il est facile de voir que la fonction objectif de tout problème d'optimisation appartenant aux deux premières classes mentionnées au chapitre II, paragraphe 3, peut prendre la forme (10).

Les problèmes de la troisième classe sont plus complexes et ne peuvent être ramenés à cette forme, vu le fait que la variable temps intervenant dans le critère est une variable discrète. Dans la suite, nous ne nous en occuperons plus.

Tenant compte des notations introduites plus haut, le problème d'optimisation peut finalement se formuler de la manière suivante :

$$(11) \quad \begin{array}{l} \text{minimiser } j(y,u) \\ \text{sous les contraintes} \\ K \quad y_t = f_t(y_t, \dots, y_{t-p}, u_t, \dots, u_{t-q}) \end{array}$$

- c) Notons d_t le vecteur de R^M

$$d_t = (y_{t-1}, y_{t-2}, \dots, y_{t-p}, u_t, u_{t-1}, \dots, u_{t-q})$$

avec $M = p \cdot N_{endo} + (q+1) \cdot N_{ins}$

Le vecteur d_t contient toute l'information connue au moment de la résolution du modèle pour la période t .

On voit donc que résoudre le modèle pour la période t , c'est résoudre le système algébrique de dimension *Nendo* :

$$(12) \quad K y_t = f_t(y_t, d_t)$$

où K , f_t et d_t sont connus.

§ 3 HYPOTHÈSES GÉNÉRALES ET VALIDITÉ DE L'ÉTUDE

La formulation standard d'un problème de contrôle optimal est la suivante : on suppose que toutes les fonctions définissant le modèle sont continûment différentiables pour toutes les valeurs possibles de leurs arguments et que, pour tout contrôle u , le modèle a une solution et une seule, notée $y(u)$. On minimise alors le critère $J(u) = j(y(u), u)$.

Ce cadre théorique est ici inapplicable. En effet, un modèle macro-économique non linéaire contient souvent des logarithmes et toujours des quotients de variables : elles ne sont continues ni même définies pour toutes les valeurs possibles de leurs arguments. Il est de plus impossible d'affirmer a priori que la solution du modèle est unique.

Remarquons cependant que le modèle macro-économique, vu la façon dont il est construit, n'est qu'une estimation du comportement réel de l'économie dans le voisinage d'un état de référence obtenu par extrapolation du passé. On peut remarquer qu'après un choc brutal sur l'économie, les modèles doivent parfois être reconstruits ou réestimés, mais que, par contre, dans une période "stable" pour laquelle le modèle reste valable, le comportement de l'économie est dynamiquement continu : la solution y_t pour la période t est "proche" de la solution y_{t-1} pour la période précédente.

Il n'est donc pas gênant, au point de vue pratique, de faire les hypothèses suivantes. On suppose donné un sous-ensemble de valeurs "raisonnables" pour les contrôles, tel que pour tout $u \in U$, le modèle (7) ait au moins une solution. Alors, pour tout $u \in U$, on appelle $y(u)$ la trajectoire définie de la manière suivante : $y_1(u)$ est la solution du système (7) à la période 1 la plus proche dans R^{Nendo} de l'état initial connu y_0 , $y_2(u)$ est la solution du système (7) à la période 2 la plus proche dans R^{Nendo} de l'état $y_1(u)$ calculé précédemment, et ainsi de suite. Dans le cas bien improbable où pour une période t plusieurs solutions de (7) seraient à égale distance de y_{t-1} , y_t serait tiré au sort parmi ces solutions. On construit ainsi une application $u \rightarrow y(u)$ qui est définie et univoque de U dans le sous-ensemble $Y = y(U)$ de R^{Nendo} , application que nous appellerons par la suite la solution du modèle.

On suppose alors que pour tout $u \in U$, tout $y \in Y$ et tout $t=1,2,\dots,T$, toutes les fonctions f_t^i et j_t définissant le modèle sont continûment différentiables, ce qui est possible car, par construction, Y est un ensemble "raisonnable" de valeurs des variables endogènes.

§ 0 CHOIX DE LA STRATÉGIE

Dans ce paragraphe nous allons proposer une stratégie pour traiter les problèmes décrits au chapitre II.

- a) On peut considérer le problème (11) comme un problème d'optimisation sous contraintes.

Dans ce contexte, il n'y a plus de distinction mathématique entre les variables endogènes et les variables de contrôle.

Pour MARIBEL dans le cas où $Nins = 10$ et $T = 10$, comme il y a 461 variables endogènes, on travaillera dans un espace de dimension $(461+10) \cdot 10 = 4710$, sous contrainte de $461 \cdot 10 = 4610$ équations non linéaires.

Actuellement les algorithmes existants ne permettent que difficilement le traitement de problèmes d'une telle taille.

- b) Une autre méthode (méthode de pénalisation) consiste à considérer les équations du modèle comme des contraintes, les variables endogènes comme des contrôles supplémentaires et à minimiser le critère $j(y,u)$ auquel on ajoute une pénalisation qui est un coût de violation des contraintes. On optimise alors sans contrainte le critère pénalisé.

Lorsque le coût de violation des contraintes augmente indéfiniment, la solution du problème pénalisé converge vers la solution du problème initial.

(Pour plus de détails et pour toute justification théorique de la méthode de pénalisation, voir <14>).

Malgré la possibilité de tirer profit de la structure creuse du modèle, les tests entrepris par Gabay et al n'ont pas donné des résultats convaincants pour cette méthode, mais toutes les possibilités n'ont pas été épuisées.

- c) Vu les hypothèses d'existence et d'unicité de la solution y du modèle en fonction de u , on peut mettre le critère $j(y,u)$ sous la forme :

$$J(u) = j(y(u), u)$$

Nous sommes donc en présence d'un problème d'optimisation non contraint dont la fonction objectif est définie implicitement à travers le modèle.

Remarquons qu'une évaluation du critère à optimiser nécessite une simulation complète du modèle et, par conséquent, est fort coûteuse. Nous avons donc intérêt à choisir un algorithme utilisant le moins d'appels de fonction possible, même si cela implique un coût de calcul plus élevé entre deux appels successifs. Suite à ces réflexions, un algorithme d'optimisation utilisant le gradient du critère semble être mieux adapté.

Comme le critère est défini implicitement par l'intermédiaire du modèle en fonction de u , nous ne connaissons sûrement pas l'expression analytique du gradient : il faudra l'approcher.

Une approximation du gradient par différence finie est peu souhaitable vu le nombre énorme d'évaluations nécessaires du critère.

Au paragraphe 6 de ce chapitre, nous exposerons la méthode de l'état adjoint, proposée par Gabay et al (<2>).

Nous verrons que cette méthode nécessite le calcul du jacobien du système d'équations. C'est entre autres pour cette raison que nous proposons un algorithme utilisant le jacobien pour la résolution du modèle (voir chapitre VII).

Pour la résolution du modèle, il sera indispensable de tenir compte du creux de celui-ci. Parmi les différentes possibilités, nous mentionnons celle de Curtis-Powell-Reid permettant un gain de temps considérable lors de l'évaluation du jacobien, et la réduction à un système quasi-triangular. Dans le paragraphe suivant, nous expliquerons notre préférence pour la réduction au système quasi-triangular.

Remarque importante :

Il n'est nullement dit que les méthodes choisies sont les meilleures. Les justifications de nos choix sont plutôt intuitives et basées essentiellement sur les tests entrepris par Gabay et al, et nous pensons qu'il est hors de propos de tester réellement toutes les méthodes possibles dans le cadre de ce travail.

Dans l'étude des modèles macro-économiques, toutes les possibilités n'ont sûrement pas été épuisées et comparées... Il y a encore de quoi occuper des mémorisants...

§ 0 ÉTUDE DE LA STRUCTURE

a) Notion de matrice d'incidence

Soit un système d'équations :

$$G(x) = 0$$

avec $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$

ou

$$(13) \quad g_i(x_1, x_2, \dots, x_n) = 0 \quad i=1, 2, \dots, m$$

avec $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$

Nous appelons matrice d'incidence E' de ce système la matrice booléenne (les éléments valent 0 ou 1) de dimension $m \times n$ d'éléments e'_{ij} tel que

$$(14) \quad e'_{ij} = \begin{cases} 1 & \text{si la variable } x_j \text{ intervient dans l'équation } g_i \\ & \text{i.e. } \frac{\partial g_i}{\partial x_j} \neq 0 \\ 0 & \text{sinon} \end{cases}$$

Pour le cas particulier d'un système macro-économique de la forme

$$k_i y_t^i = f_t^i(y_t, d_t) \quad i=1, 2, \dots, Nendo \quad (12)$$

la définition de la matrice d'incidence E de dimension $Nendo \times Nendo$ est légèrement différente :

$$(15) \quad \text{pour } i \neq j : e_{ij} = \begin{cases} 1 & \text{si la variable } y_t^i \text{ intervient dans} \\ & \text{l'équation } f_t^i \text{ i.e. } \frac{\partial f_t^i}{\partial y_t^i} \neq 0 \\ 0 & \text{sinon} \end{cases}$$

$$e_{ii} = \begin{cases} 0 & \text{si } k_i = 1 \\ 1 & \text{si } k_i = 0 \end{cases}$$

Il est clair qu'en pratique nous ne recalculerons pas la matrice d'incidence E définie en (15) pour chaque simulation.

D'autre part E dépend de la correspondance biunivoque entre les variables endogènes et les équations, correspondance propre à chaque simulation (voir paragraphe 1 de ce chapitre). Lors d'une exogénéisation ou d'une endogénéisation le nombre N_{endo} des variables endogènes peut même varier, donc la dimension de la matrice d'incidence E n'est pas déterminée a priori.

Nous calculerons une fois pour toutes la matrice d'incidence E' définie en (14) des N_{equa} équations se trouvant dans MODEL par rapport aux N variables (endogènes et exogènes) considérées être sous la forme (13) dans laquelle $n=N$ et $m=N_{\text{equa}}$.

A l'aide d'un vecteur intermédiaire appelé $IRAV$ et de la sous-routine ISE , nous tirerons l'information nécessaire pour E de E' , sans réellement stocker E .

$IRAV$ dépend de chaque problème particulier et est construit à partir de $IDVAR$ de la façon suivante :

$$(16) \quad IRAV(I) = \begin{cases} J & \text{si } IDVAR(J) = I \\ -J & \text{si } IDVAR(J) = -I \\ 0 & \text{si l'équation } I \text{ est supprimée} \end{cases}$$

La construction montre que $IRAV(I) = \pm J$ fait correspondre la variable J à l'équation I .

Pour $I = 1, 2, \dots, N_{\text{equa}}$

$J = 1, 2, \dots, N_{\text{equa}}$

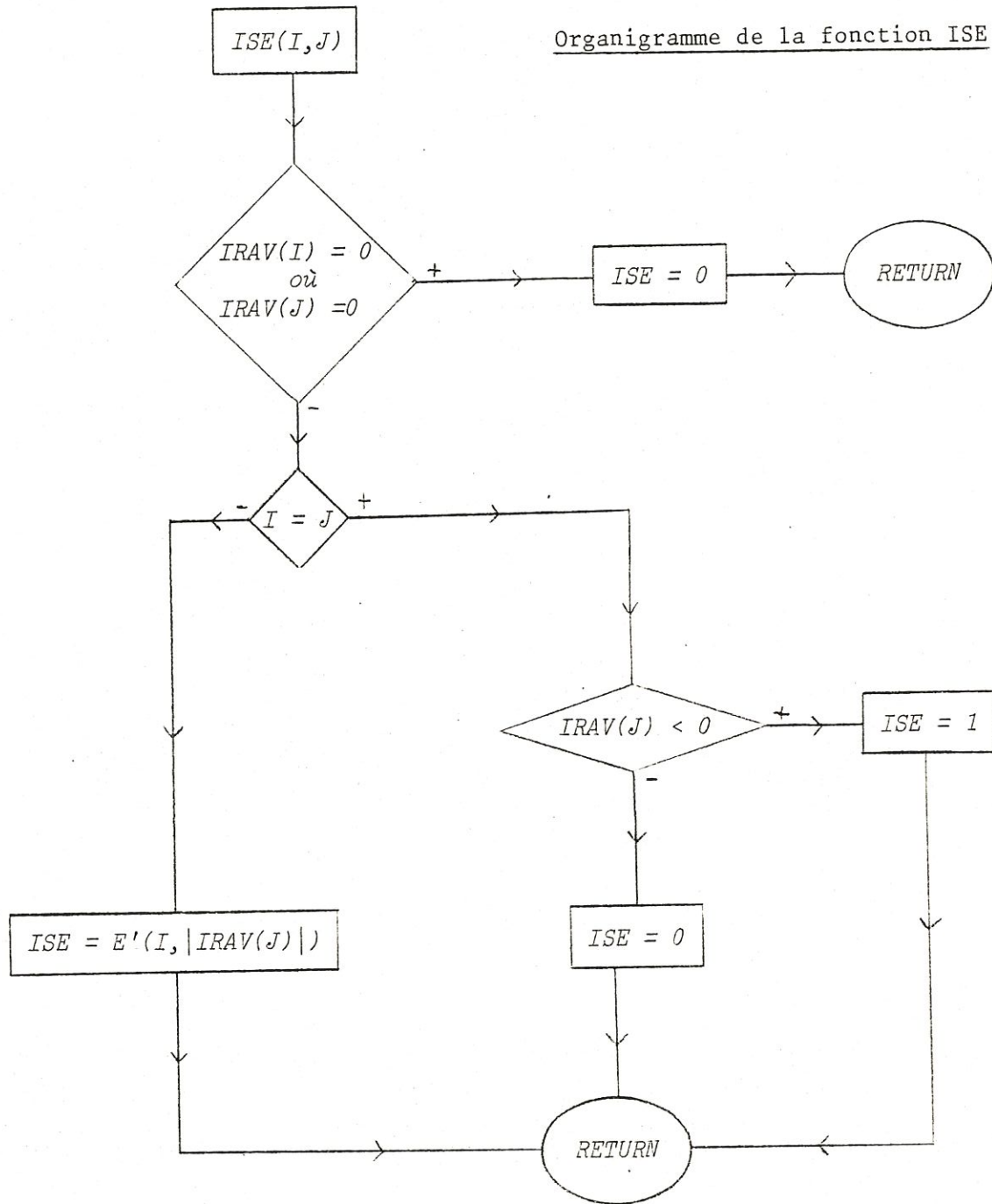
$IRAV(I) \neq 0$

$IRAV(J) \neq 0$

nous avons :

pour $I \neq J \quad E(I, J) = E'(I, |IRAV(J)|)$

$$E(I, I) = \begin{cases} 1 & \text{si } IRAV(I) < 0 \text{ (car } k_i = 0) \\ 0 & \text{si } IRAV(I) > 0 \text{ (car } k_i = 1) \end{cases}$$



Remarquons que nous travaillons en réalité sur une matrice d'incidence E fictive (car elle n'est stockée nulle part) de dimension $Nendo \times Nendo$. Les lignes correspondant aux équations supprimées, ainsi que les colonnes correspondant aux variables exogénéisées seront considérées comme nulles.

La figure 4 reproduit l'organigramme de la fonction ISE qui retourne la valeur de $E(I,J)$.

Reprenons encore l'exemple du modèle TOTO et construisons d'abord la matrice d'incidence E' définie en (14) :

$$E' = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

Comme le vecteur $IRAV$ est construit à partir du vecteur $IDVAR$ et comme celui-ci dépend de chaque problème particulier, le vecteur $IRAV$ ainsi que la matrice d'incidence E définie en (15) (dont les éléments sont évalués à l'aide de $IRAV$) varient suivant le problème posé.

Envisageons d'abord le cas initial (I) de TOTO.

Pour ce problème, le vecteur $IRAV$ défini en (16) s'écrit :

$$IRAV(1) = 1$$

$$IRAV(2) = 2$$

$$IRAV(3) = 4$$

$$IRAV(4) = 7$$

Remarquons que les vecteurs $IRAV$ et $IWATEQ$ sont égaux pour ce problème particulier. En effet, ici la variable correspondant à une équation donnée est toujours sa variable d'assignation.

Construisons maintenant la matrice E .

La colonne I de E est en fait la colonne $IRAV(I)$ de E' , à l'élément diagonal près.

Nous obtenons donc :

$$\begin{array}{lcl}
 \text{colonne de } E' & 1 & 2 \ 4 \ 7 \\
 \text{colonne de } E & 1 & 2 \ 3 \ 4 \\
 E = & \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}
 \end{array}$$

↓

$$\text{colonne IRAV}(3) = 4$$

Les éléments diagonaux sont tous nuls car les k_i valent tous 1.

Le système

$$\begin{cases}
 y_1 = g_1(y_3, y_5) \\
 y_2 = g_2(y_1, y_4, y_5, y_6, y_7) \\
 y_4 = g_3(y_1, y_2, y_5, y_6) \\
 y_7 = g_4(y_3, y_4)
 \end{cases} \quad (4)$$

peut se réécrire

$$\begin{cases}
 y_1 = g_1 \\
 y_2 = g_2(y_1, y_4, y_7) \\
 y_4 = g_3(y_1, y_2) \\
 y_7 = g_4(y_4)
 \end{cases}$$

en considérant les variables endogènes y_3, y_5, y_6 comme des constantes définissant les fonctions g_i .

Ou encore :

$$\begin{cases}
 y_{\text{IRAV}(1)} = g_1 \\
 y_{\text{IRAV}(2)} = g_2(y_{\text{IRAV}(1)}, y_{\text{IRAV}(3)}, y_{\text{IRAV}(4)}) \\
 y_{\text{IRAV}(3)} = g_3(y_{\text{IRAV}(1)}, y_{\text{IRAV}(2)}) \\
 y_{\text{IRAV}(4)} = g_4(y_{\text{IRAV}(3)})
 \end{cases}$$

Ou encore, en notant abusivement $y_{\text{IRAV}(i)} = y_i$

$$(17) \quad \begin{cases} k_1 y_1 = f_1 \\ k_2 y_2 = f_2(y_1, y_3, y_4) \\ k_3 y_3 = f_3(y_1, y_2) \\ k_4 y_4 = f_4(y_3) \end{cases}$$

où $f_i = g_i$ pour $i=1, \dots, 4$
 $k_i = 1$

Ainsi nous avons ramené le système à la forme (4).

- Reprenons maintenant le cas II 2)b) pour lequel IDVAR vaut :

$$\begin{aligned} \text{IDVAR}(1) &= 1 \\ \text{IDVAR}(2) &= 2 \\ \text{IDVAR}(3) &= 2000 \\ \text{IDVAR}(4) &= 2000 \\ \text{IDVAR}(5) &= 2000 \\ \text{IDVAR}(6) &= 3 \\ \text{IDVAR}(7) &= 4 \end{aligned}$$

On peut en tirer immédiatement les valeurs pour IRAY :

$$\begin{aligned} \text{IRAY}(1) &= 1 \\ \text{IRAY}(2) &= 2 \\ \text{IRAY}(3) &= 6 \\ \text{IRAY}(4) &= 7 \end{aligned}$$

Nous pouvons obtenir E de la même façon que précédemment :

$$\begin{array}{l} \text{colonnes de } E' \quad 1 \ 2 \ 6 \ 7 \\ \text{colonnes de } E \quad 1 \ 2 \ 3 \ 4 \\ E = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

- En dernier lieu, envisageons le cas II 3)b)

où

$$\begin{aligned} IDVAR(1) &= 1 \\ IDVAR(2) &= 2 \\ IDVAR(3) &= 2000 \\ IDVAR(4) &= 2000 \\ IDVAR(5) &= 2000 \\ IDVAR(6) &= 2000 \\ IDVAR(7) &= -3 \end{aligned}$$

Nous en déduisons IRAY :

$$\begin{aligned} IRAY(1) &= 1 \\ IRAY(2) &= 2 \\ IRAY(3) &= -7 \\ IRAY(4) &= 0 \end{aligned}$$

Comme l'équation 4 est supprimée, la ligne ainsi que la colonne 4 de la matrice E sont mises à zéro.

L'élément diagonal e_{33} vaut 1 dans ce cas-ci, car $k_3 = 0$ (parce que IRAY(3) est négatif).

Voici la matrice d'incidence E correspondant à ce problème :

$$E = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

b) Méthode CPR pour l'évaluation du jacobien

Considérons le système d'équations (13) et sa matrice d'incidence définie en (14).

L'approximation par différence finie de la matrice jacobienne

$$J = \left[\frac{\partial g_i}{\partial x_j} \right] \begin{matrix} i=1, \dots, m \\ j=1, \dots, n \end{matrix}$$

peut être calculée de deux manières différentes :

$$(18) \quad \frac{\partial g_i}{\partial x_j} = \frac{g_i(x_1, \dots, x_{j-1}, x_j+h_j, x_{j+1}, \dots, x_n) - g_i(x_1, \dots, x_j, \dots, x_n)}{h_j}$$

$$(19) \quad \frac{\partial g_i}{\partial x_j} = \frac{g_i(x_1+h_1, \dots, x_j+h_j, x_{j+1}, \dots, x_n) - g_i(x_1+h_1, \dots, x_{j-1}+h_{j-1}, x_j, \dots, x_n)}{h_j}$$

Sans étude de la structure ce calcul nécessite $(n+1)$ évaluations du système entier.

Curtis-Powell-Reid proposent une méthode qui tient compte de la structure et qui offre ainsi un gain de temps calcul considérable pour des matrices creuses.

L'idée est la suivante :

En utilisant la formule (19) à une renumérotation près et en tenant compte de la notation suivante :

$$g_i^{kl} = g_i(x_1, \dots, x_{k-1}, x_k+h_k, x_{k+1}, \dots, x_{l-1}, x_l+h_l, x_{l+1}, \dots, x_n)$$

nous avons :

$$\begin{aligned} \frac{\partial g_i}{\partial x_k} h_k + \frac{\partial g_i}{\partial x_l} h_l &= g_i^{kl} - g_i^l + g_i^l - g_i \\ (20) \quad \frac{\partial g_i}{\partial x_k} h_k + \frac{\partial g_i}{\partial x_l} h_l &= g_i^{kl} - g_i \end{aligned}$$

Supposons que la colonne k et la colonne l de la matrice d'incidence E' n'aient jamais un 1 sur la même ligne, c'est-à-dire

$$E'(I, K) = 0 \quad \text{ou} \quad E'(I, L) = 0$$

Sachant qu'une des deux dérivées partielles est nulle, nous pouvons immédiatement calculer l'autre par la formule (20).

Nous n'avons donc fait qu'une évaluation du système $(g_i^{kl}, i=1, \dots, m)$ pour calculer deux colonnes de la matrice jacobienne.

Ce raisonnement s'étend facilement à plusieurs colonnes.

Pour une matrice fort creuse le gain d'évaluations peut donc être élevé, mais le nombre maximum d'éléments non nuls sur une ligne de la matrice d'incidence constitue une borne inférieure pour le nombre d'évaluations de fonction.

Remarquons encore que dans le cas de MARIBEL cette borne vaut 30.

c) Réduction à la forme quasi-triangulaire

Soit le modèle macro-économique (10) avec sa matrice d'incidence E.

Supposons que E soit strictement triangulaire inférieure, c'est-à-dire que

$$e_{ij} = 0 \quad \forall j \geq i$$

Le système d'équations est alors récursif, c'est-à-dire qu'on peut tirer y_1 de la première équation, connaissant y_1 on peut calculer y_2 dans la seconde, et ainsi de suite.

Supposons que dans la colonne l de E il y ait des éléments non nuls au-dessus de la diagonale. Il est facile de voir qu'en fixant la valeur de la variable y_l , le système devient récursif. Une telle variable est appelée variable de bouclage.

Mathématiquement, y_j est une variable de bouclage

$$\text{ssi} \quad \prod_{i=1}^j (1 - e_{ij}) = 0$$

$$\text{ou ssi} \quad \prod_{i=1}^j (1 - e_{ij}) = 0$$

Une équation correspondant à une variable de bouclage est appelée équation de bouclage.

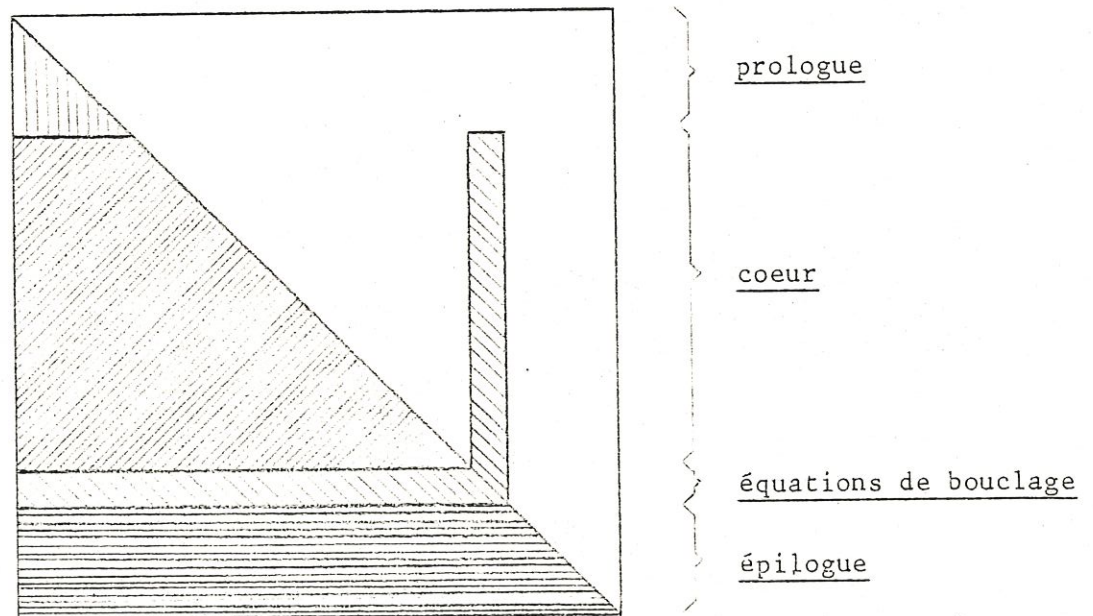
Il est important de constater que la structure de la matrice d'incidence et donc aussi le nombre de variables de bouclage dépendent de l'ordre des équations.

Dans une matrice d'incidence on peut distinguer trois parties :

- une partie réursive de départ
- une partie contenant des variables de bouclage yb
- une partie qui devient réursive dès que les équations correspondant aux deux premières sont résolues.

Par une renumérotation on mettra les équations constituant la première partie au début, elles seront suivies de celles de la deuxième, réordonnées de façon à ce que les équations de bouclage se trouvent à la fin, suivies des équations de la troisième partie.

Après renumérotation, la matrice d'incidence aura la forme suivante, appelée forme quasi-triangulaire :



- figure 5 -

Les différentes parties sont appelées respectivement prologue, coeur, équations de bouclage et épilogue. Dans la suite nous supposons que les variables du prologue sont numérotées de 1 à l, celles du coeur de l+1 à m, qu'il y a s variables de bouclage numérotées de m+1 à m+s et que les numéros des variables de l'épilogue varient de m+s+1 à Nequa.

Compte tenu de la structure de la matrice d'incidence renumérotée et du fait que k_i vaut zéro pour les équations du prologue, du coeur et de l'épilogue, le modèle peut donc s'écrire :

(22)	<u>Prologue</u> :	$y_t^i = f_t^i(y_t^1, y_t^2, \dots, y_t^{i-1}, d_t)$	$i=1, \dots, l$
(23)	<u>Coeur</u> :	$y^i = f_t^i(y_t^1, y_t^2, \dots, y_t^{i-1}, y_{b_t}, d_t)$	$i=l+1, \dots, m$
(24)	<u>Equations de bouclage</u> :	$k_i y^i = f_t^i(y_t^1, y_t^2, \dots, y_t^{m+s}, d_t)$	$i=m+1, \dots, m+s$
(25)	<u>Epilogue</u> :	$y_t^i = f_t^i(y_t^1, y_t^2, \dots, y_t^{i-1}, d_t)$	$i=m+s+1, \dots, Nequa$

Remarquons que les différents blocs peuvent être vides.

Comment peut-on profiter de cette structure pour la résolution ?

- Le prologue peut être résolu directement, indépendamment des autres équations.
- Nous avons vu que le coeur devient récursif dès qu'on fixe les valeurs des variables de bouclage y_b . Le problème consiste à trouver les bonnes valeurs de ces variables, c'est-à-dire celles qui vérifient les équations de bouclage. Ceci donne l'idée d'itérer sur les variables de bouclage. Ils'agit donc de résoudre le système des s équations de bouclage définies implicitement par l'intermédiaire du coeur i.e.

$$E_t(y_{b_t}) = 0$$

où

$$(26) \quad E_t^i(y_{b_t}) = f_{b_t}^i(y_t, d_t) - l_i y_{b_t}^i$$

où

$$(27) \quad E_t(yb_t) = fb_t(y_t, d_t) - L yb_t$$

avec

$$yb_t^i = y_t^{m+i} \quad (i^{i\text{ème}} \text{ variable de bouclage})$$

$$fb_t^i = f_t^{m+i}$$

$$l_i = k_{m+i}$$

$$L = \text{diag.}(l_i)$$

Ceci réduit la taille du système à résoudre de $m+s-1$ à s équations, pour MARIBEL de 300 à 18.

- Il ne reste plus qu'à résoudre le système récursif de l'épilogue.

Dans le chapitre suivant, nous allons exposer des méthodes pour réduire la matrice d'incidence à la forme quasi-triangulaire. Il est évidemment intéressant que le système à résoudre soit de dimension peu élevé. Cependant, *il n'est nullement dit que le nombre de variables de bouclage minimal entraîne un temps de calcul minimal*, étant donné que toute la non-linéarité du coeur passe dans le système de bouclage.

La mise au point d'un algorithme de résolution d'un système d'équations non linéaires se fera au chapitre VII.

Au chapitre V nous montrerons comment profiter du creux dans la cheminée (voir figure 5) pour le calcul du jacobien du système de bouclage nécessaire à la résolution.

Cependant, nous allons déjà dégager l'expression analytique de celui-ci en fonction du jacobien du système entier.

En notant yc_t et fc_t respectivement les variables et équations du coeur, (23) donne :

$$(28) \quad \frac{\partial yc_t}{\partial yb_t} = \frac{\partial fc_t}{\partial yc_t} \cdot \frac{\partial yc_t}{\partial yb_t} + \frac{\partial fc_t}{\partial yb_t}$$

La dérivation de l'équation (27) donne :

$$(29) \quad \frac{\partial E_t}{\partial y_b_t} = \frac{\partial f_b_t}{\partial y_c_t} \cdot \frac{\partial y_c_t}{\partial y_b_t} + \frac{\partial f_b_t}{\partial y_b_t} - L$$

Da la formule (28) nous tirons :

$$(30) \quad \frac{\partial y_c_t}{\partial y_b_t} = \left(I - \frac{\partial f_c_t}{\partial y_c_t} \right)^{-1} \cdot \frac{\partial f_c_t}{\partial y_b_t}$$

En remplaçant (30) dans (29), nous obtenons :

$$(31) \quad \frac{\partial E_t}{\partial y_b_t} = \frac{\partial f_b_t}{\partial y_c_t} \cdot \left(I - \frac{\partial f_c_t}{\partial y_c_t} \right)^{-1} \cdot \frac{\partial f_c_t}{\partial y_b_t} + \frac{\partial f_b_t}{\partial y_b_t} - L$$

§ 0 MÉTHODE DE L'ÉTAT ADJOINT

Nous avons déjà évoqué au paragraphe 4 de ce chapitre que le calcul du gradient du critère

$$J(u) = j(y(u), u)$$

par différence finie est fort coûteux. En effet, il faut évaluer le critère pour $N_{ins} \cdot T + 1 = Ct+1$ valeurs différentes du contrôle u (car il faut donner successivement un accroissement à chaque composante de u , voir formules (18) et (19)), c'est-à-dire faire $Ct+1$ simulations du modèle.

La méthode de l'état adjoint, proposée par Gabay et al (voir <2>), permet d'une manière moins coûteuse l'approximation du gradient $\nabla J(u)$ en question. Elle est basée sur la théorie des équations adjointes du contrôle optimal (voir <11>).

Soit δ_t l'indicateur défini par :

$$\delta_t = \begin{cases} 1 & \text{si } t \leq T \\ 0 & \text{si } t > T \end{cases}$$

Soit $z_t \in R^{N_{endo}}$ et $w_t \in R^{N_{ins}}$ les vecteurs calculés à partir des gradients en y et en u du critère $j(y,u)$ défini en (10) :

$$(33a) \quad z_t = \sum_{k=0}^p \delta_{t+k} \frac{\partial j_{t+k}}{\partial y_t} \quad \text{pour } t=1,2,\dots,T$$

$$(33b) \quad w_t = \sum_{k=0}^q \delta_{t+k} \frac{\partial j_{t+k}}{\partial u_t}$$

Enfin, soit $\Psi \in R^{Mt}$ la solution du système

$$(34) \quad K \Psi_t = z_t + \sum_{k=0}^p \delta_{t+k} \left(\frac{\partial f_{t+k}}{\partial y_t} \right)^* \Psi_{t+k} \quad \text{pour } t=1,2,\dots,T$$

où A^* désigne la matrice transposée de A ,

K est la matrice définie en (9)

et $\Psi = (\Psi_1, \Psi_2, \dots, \Psi_t)$

Le système (34) est appelé équation adjointe et sa solution Ψ état adjoint.

On notera que l'état adjoint dépend à la fois du modèle par l'intermédiaire de $\left(\frac{\partial f_{t+k}}{\partial y_t} \right)$ et du critère par z_t .

Théorème

Supposons que les fonctions f_t et j_t définissant le modèle et le critère soient continûment différentiables pour tout $u \in U$ et $y \in y(U)$ (définis au paragraphe 3 de ce chapitre). Soit un u quelconque de U et $y(u)$ la solution correspondante du modèle. Supposons que les vecteurs z_t, w_t et les matrices $\frac{\partial f_{t+k}}{\partial y_t}$ et $\frac{\partial f_{t+k}}{\partial u_t}$ soient calculés pour le couple $(u, y(u))$.

Alors le gradient $\nabla J(u)$ du critère $J(u) = j(y(u), u)$ est donné par

$$(35) \quad \nabla J(u)_t = w_t + \sum_{k=0}^q \delta_{t+k} \left(\frac{\partial f_{t+k}}{\partial u_t} \right)^* \psi_{t+k} \quad \text{pour } t=1, 2, \dots, T$$

Rappel

Soit S un ouvert non vide de \mathbb{R}^n

Soit $f : S \rightarrow \mathbb{R}$

Soit $\bar{x} \in S$

f est dite différentiable en \bar{x}

ssi

un vecteur, noté $\nabla f(\bar{x}) \in \mathbb{R}^n$, appelé gradient de f au point \bar{x}

et une fonction $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$

tel que

$$f(x) = f(\bar{x}) + (\nabla f(\bar{x}), (x - \bar{x})) + \|x - \bar{x}\| \alpha(x - \bar{x}) \quad \forall x \in S$$

où

$$\lim_{x \rightarrow \bar{x}} \alpha(x - \bar{x}) = 0$$

ou encore

$$(36) \quad \frac{f(\bar{x} + hv) - f(\bar{x})}{h} = (\nabla f(\bar{x}), v) + \theta(h) \quad \begin{array}{l} \forall v \in \mathbb{R}^n \\ h \in \mathbb{R} \\ \text{avec } \bar{x} + hv \in S \end{array}$$

où $(.,.)$ désigne le produit scalaire.

Démonstration

Soit v un élément quelconque de R^{Nc} .

Soit h un scalaire suffisamment petit pour que $u+hv \in U$.

On pose successivement :

$$\begin{aligned} u_h &= u + hv \\ y &= y(u) \\ y_h &= y(u_h) \\ \tilde{y} &= \frac{y_h - y}{h} \\ \Delta_h J &= \frac{J(u_h) - J(u)}{h} \end{aligned}$$

De la définition du critère $J(u) = j(y(u), u)$, on déduit :

$$(38) \quad \Delta_h J = \frac{j(y_h, u_h) - j(y, u)}{h}$$

D'où, d'après la définition (10) du critère j :

$$(39) \quad \Delta_h J = \sum_{t=1}^T \left(\sum_{k=0}^p \left(\frac{\partial j_t}{\partial y_{t-k}}, \tilde{y}_{t-k} \right) + \sum_{k=0}^q \left(\frac{\partial j_t}{\partial u_{t-k}}, v_{t-k} \right) \right) + \theta(h)$$

Il est clair que \tilde{y}_t et v_t sont nuls pour $t \leq 0$.

Utilisant les notations (32) et (33), on vérifiera que (39) est équivalent à :

$$(40) \quad \Delta_h J = \sum_{t=1}^T \left((z_t, \tilde{y}_t) + (w_t, v_t) \right) + \theta(h)$$

Tirons z_t du système adjoint (34) et portons le dans (40).

On obtient :

$$(41) \quad \Delta_h J = \sum_{t=1}^T (K \psi_t, \tilde{y}_t) - \sum_{t=1}^T \sum_{k=0}^p \left(\left(\frac{\partial f_{t+k}}{\partial y_t} \right)^* \psi_{t+k}, \tilde{y}_t \right) \delta_{t+k} \\ + \sum_{t=1}^T (w_t, v_t) + \theta(h)$$

En utilisant la propriété bien connue du produit scalaire

$$(42) \quad (Ax, y) = (x, A^* y)$$

et le fait que \tilde{y}_t est nul pour tout $t \leq 0$,

(41) est encore équivalent à :

$$(43) \quad \Delta_h J = \sum_{t=1}^T \left(\psi_t, K \tilde{y}_t - \sum_{k=0}^p \frac{\partial f_t}{\partial y_{t-k}} \tilde{y}_{t-k} \right) + \sum_{t=1}^T (w_t, v_t) + \theta(h)$$

Par ailleurs la dérivation du modèle (7) nous donne, avec la notation (9) :

$$(44) \quad K \tilde{y}_t = \sum_{k=0}^p \frac{\partial f_t}{\partial y_{t-k}} \tilde{y}_{t-k} + \sum_{k=0}^q \frac{\partial f_t}{\partial u_{t-k}} v_{t-k} + \theta(h)$$

Portant (44) dans (43), on obtient :

$$(45) \quad \Delta_h J = \sum_{t=1}^T \sum_{k=0}^q \left(\psi_t, \frac{\partial f_t}{\partial u_{t-k}} v_{t-k} \right) + \sum_{t=1}^T (w_t, v_t) + \theta(h)$$

En utilisant encore la propriété (42) et en tenant compte du fait que v_t est nul pour tout $t \leq 0$, on voit que (45) est encore équivalent à :

$$(46) \quad \Delta_h J = \sum_{t=1}^T \left(w_t + \sum_{k=0}^q \delta_{t+k} \left(\frac{\partial f_{t+k}}{\partial u_t} \right)^* \psi_{t+k}, v_t \right) + \theta(h)$$

Le rappel (36) nous permet d'écrire :

$$(47) \quad \lim_{h \rightarrow 0} \Delta_h J = \lim_{h \rightarrow 0} \frac{J(u_h) - J(u)}{h} = (\nabla J(u) , v)$$

En passant à la limite dans la formule (46), on obtient :

$$(48) \quad (\nabla J(u)_t , v_t) = \sum_{t=1}^T (w_t + \sum_{k=0}^q \delta_{t+k} \left(\frac{\partial f_{t+k}}{\partial u_t} \right)^* \Psi_{t+k} , v_t)$$

Le résultat (48) ayant été obtenu pour un v quelconque, le théorème est démontré. ■

Montrons maintenant comment on résoud le système adjoint (34) :

$$K \Psi_t = z_t + \sum_{k=0}^p \delta_{t+k} \left(\frac{\partial f_{t+k}}{\partial y_t} \right)^* \Psi_{t+k} \quad (34)$$

Pour $t=T$, le système prend la forme :

$$(49) \quad K \Psi_T = z_T + \left(\frac{\partial f_T}{\partial y_T} \right)^* \Psi_T$$

ce qui permet de calculer Ψ_T .

De même pour $t=T-1$, (34) s'écrit :

$$(50) \quad K \Psi_{T-1} = z_{T-1} + \left(\frac{\partial f_{T-1}}{\partial y_{T-1}} \right)^* \Psi_{T-1} + \left(\frac{\partial f_T}{\partial y_{T-1}} \right)^* \Psi_T$$

ce qui connaissant Ψ_T , permet de calculer Ψ_{T-1} , et ainsi de suite.

La résolution de l'équation adjointe se ramène donc à la résolution de T systèmes linéaires de dimension N_{endo} que l'on résoud par rétrogradation, c'est-à-dire pour $t=T, T-1, \dots, 1$.

Notons

$$(51) \quad \hat{z}_t = z_t + \sum_{k=1}^p \delta_{t+k} \left(\frac{\partial f_{t+k}}{\partial y_t} \right)^* \psi_{t+k} \quad \text{pour } t=1, 2, \dots, T$$

La résolution de (34) se ramène à la résolution des T systèmes linéaires :

$$(52) \quad K \psi_t = \hat{z}_t + \left(\frac{\partial f_t}{\partial y_t} \right)^* \psi_t \quad \text{pour } t=1, 2, \dots, T$$

où \hat{z}_t est connu.

Pour résoudre le système (52) avec t fixé, nous allons à nouveau tenir compte de la structure quasi-triangulaire, étant donné que la matrice $\left(\frac{\partial f_t}{\partial y_t} \right)$ a la même structure que la matrice d'incidence du modèle.

Dans la suite, nous omettrons parfois l'indice t (dans le cas où t est fixé), pour faciliter les écritures.

La matrice $\left(\frac{\partial f_t}{\partial y_t} \right)$ peut se décomposer en 16 blocs (voir figure 6).

Dans la suite, fp , fc , fb , fe désignent respectivement les équations du prologue, du coeur, de bouclage, de l'épilogue, et yp , yc , yb , ye les variables correspondantes.

Remarquons que les blocs suivants sont nuls :

$$\begin{aligned} & \frac{\partial fp}{\partial yc}^*, \frac{\partial fp}{\partial yb}^*, \frac{\partial fp}{\partial ye}^* \\ & \frac{\partial fc}{\partial ye}^* \\ & \frac{\partial fb}{\partial ye}^* \end{aligned}$$

et que les blocs

$$\frac{\partial f_p}{\partial y_p}^*, \frac{\partial f_c}{\partial y_c}^* \text{ et } \frac{\partial f_e}{\partial y_e}^*$$

sont strictement triangulaires supérieurs

Décomposition en blocs de la matrice $\left(\frac{\partial f_t}{\partial y_t} \right)^*$

$\frac{\partial f_p}{\partial y_p}^*$	$\frac{\partial f_c}{\partial y_p}^*$	$\frac{\partial f_b}{\partial y_p}^*$	$\frac{\partial f_e}{\partial y_p}^*$
$\frac{\partial f_p}{\partial y_c}^*$ 0	$\frac{\partial f_c}{\partial y_c}^*$	$\frac{\partial f_b}{\partial y_c}^*$	$\frac{\partial f_e}{\partial y_c}^*$
$\frac{\partial f_p}{\partial y_b}^*$ 0	$\frac{\partial f_c}{\partial y_b}^*$	$\frac{\partial f_b}{\partial y_b}^*$	$\frac{\partial f_e}{\partial y_b}^*$
$\frac{\partial f_p}{\partial y_e}^*$ 0	0 $\frac{\partial f_c}{\partial y_e}^*$	$\frac{\partial f_b}{\partial y_e}^*$ 0	$\frac{\partial f_e}{\partial y_e}^*$

- figure 6 -

Pour t fixé, le système (52) peut alors s'écrire :

$$\begin{aligned}
 (53p) \quad & \Psi_p = \hat{z}_p + \left(\frac{\partial f_p}{\partial y_p} \right)^* \Psi_p + \left(\frac{\partial f_c}{\partial y_p} \right)^* \Psi_c + \left(\frac{\partial f_b}{\partial y_p} \right)^* \Psi_b + \left(\frac{\partial f_e}{\partial y_p} \right)^* \Psi_e \\
 (53c) \quad & \Psi_c = \hat{z}_c + \left(\frac{\partial f_c}{\partial y_c} \right)^* \Psi_c + \left(\frac{\partial f_b}{\partial y_c} \right)^* \Psi_b + \left(\frac{\partial f_e}{\partial y_c} \right)^* \Psi_e \\
 (53b) \quad & L \Psi_b = \hat{z}_b + \left(\frac{\partial f_c}{\partial y_b} \right)^* \Psi_c + \left(\frac{\partial f_b}{\partial y_b} \right)^* \Psi_b + \left(\frac{\partial f_e}{\partial y_b} \right)^* \Psi_e \\
 (53e) \quad & \Psi_e = \hat{z}_e + \left(\frac{\partial f_e}{\partial y_e} \right)^* \Psi_e
 \end{aligned}$$

Pour la résolution de (53), on commencera par tirer Ψ_e du système récursif (53e). Connaissant Ψ_c , Ψ_b et Ψ_e , il sera facile de déterminer Ψ_p en résolvant le système (53p) également récursif.

Voyons maintenant comment déterminer Ψ_c et Ψ_b .

(53c) et (53b) peuvent s'écrire :

$$\begin{aligned}
 (54c) \quad & \Psi_c = \hat{z}_c + \left(\frac{\partial f_c}{\partial y_c} \right)^* \Psi_c + \left(\frac{\partial f_b}{\partial y_c} \right)^* \Psi_b \\
 (54b) \quad & L \Psi_b = \hat{z}_b + \left(\frac{\partial f_c}{\partial y_b} \right)^* \Psi_c + \left(\frac{\partial f_b}{\partial y_b} \right)^* \Psi_b
 \end{aligned}$$

où

$$\begin{aligned}
 \hat{z}_c &= \hat{z}_c + \left(\frac{\partial f_e}{\partial y_c} \right)^* \Psi_e \\
 \hat{z}_b &= \hat{z}_b + \left(\frac{\partial f_e}{\partial y_b} \right)^* \Psi_e
 \end{aligned}$$

La matrice $\left(\frac{\partial f_c}{\partial y_c} \right)^*$ étant strictement triangulaire supérieure, pour tout Ψ_b donné le système (54c) est récursif en Ψ_c .

Plus précisément, pour tout ψ_b on sait calculer ψ_c à l'aide de (54c), et ensuite déduire une erreur, notée $\phi(\psi_b)$, sur (54b).

Il suffit donc de calculer ψ_b tel que $\phi(\psi_b) = 0$.

Il est clair que l'application ϕ est affine, donc il existe A et b tel que

$$\phi(\psi_b) = A \psi_b + b$$

Déterminons l'expression analytique de A.

Comme le déterminant d'une matrice triangulaire est le produit des éléments diagonaux, la matrice

$$\left[I - \left(\frac{\partial f_c}{\partial y_c} \right)^* \right]$$

est inversible.

(54c) donne :

$$(55) \quad \psi_c = \left[I - \left(\frac{\partial f_c}{\partial y_c} \right)^* \right]^{-1} \cdot \left[z_c + \left(\frac{\partial f_b}{\partial y_c} \right)^* \psi_b \right]$$

Portons (55) dans (54b) :

$$(56) \quad L \psi_b = z_b + \left(\frac{\partial f_c}{\partial y_c} \right)^* \cdot \left[I - \left(\frac{\partial f_c}{\partial y_c} \right)^* \right]^{-1} \cdot \left[z_c + \left(\frac{\partial f_b}{\partial y_c} \right)^* \psi_b \right] + \left(\frac{\partial f_b}{\partial y_b} \right)^* \psi_b$$

ou encore :

$$(57) \quad \phi(\psi_b) = \left[\left(\frac{\partial f_c}{\partial y_b} \right)^* \cdot \left[I - \left(\frac{\partial f_c}{\partial y_c} \right)^* \right]^{-1} \cdot \left(\frac{\partial f_b}{\partial y_c} \right)^* + \left(\frac{\partial f_b}{\partial y_b} \right)^* - L \right] \psi_b + z_b + \left(\frac{\partial f_c}{\partial y_b} \right)^* \cdot \left[I - \left(\frac{\partial f_c}{\partial y_c} \right)^* \right]^{-1} z_c$$

En comparant (31) et (57), on constate que la matrice A n'est rien d'autre que la transposée du jacobien du système de bouclage, utilisé dans l'algorithme de résolution.

Le vecteur b peut s'obtenir en calculant l'expression

$$(58) \quad b = \bar{z}b + \left(\frac{\partial f_c}{\partial y_b} \right)^* \left[I - \left(\frac{\partial f_c}{\partial y_c} \right)^* \right]^{-1} \bar{z}c$$

ou encore, en évaluant Φ pour $b = 0$.

En pratique, on utilisera la deuxième manière pour obtenir b .

Pour obtenir les valeurs de Ψ_c et Ψ_b , on résoudra donc l'équation

$$\Phi(\Psi b) = 0$$

pour remplacer ensuite Ψb par sa valeur dans (54c) et déterminer ainsi la valeur de Ψ_c .

En résumé, pour résoudre le système (52), il faudra :

- résoudre un système récursif de dimension $Nendo-(m+s)$ afin de déterminer Ψ_e
 - résoudre un système récursif de dimension $m-1$ pour évaluer $b = \Phi(0)$
 - résoudre un système linéaire de dimension s ($\Phi(\Psi b) = 0$) pour obtenir Ψb
 - évaluer Ψ_c en résolvant le système récursif de dimension $m-1$
 - déterminer Ψ_p par un système récursif de dimension 1
- et ceci pour $t=1,2,\dots,T$.

Pour la résolution de (52), la détermination de \hat{z}_t et l'évaluation de $\nabla J(u)$ par la formule (35), il faut calculer et stocker les matrices :

$$(59a) \quad \left(\frac{\partial f_{t+k}}{\partial y_t} \right) \quad \text{pour } k=0,1,\dots,p$$

et pour $t=1,2,\dots,T$

$$(59b) \quad \left(\frac{\partial f_{t+k}}{\partial u_t} \right) \quad \text{pour } k=0,1,\dots,q$$

Pour les matrices $\begin{pmatrix} \frac{\partial f_t}{\partial y_t} \end{pmatrix}$, il ne faut plus calculer le bloc $\begin{pmatrix} \frac{\partial f b_t}{\partial y b_t} \end{pmatrix}$ étant donné que cette information se trouve implicitement dans la matrice A.

Pour évaluer les éléments nécessaires de toutes ces matrices par différence finie, on pourra utiliser la méthode CPR afin d'épargner du temps calcul.

Il est évidemment impossible, même en tenant compte du creux de ces matrices, de stocker celles-ci en mémoire centrale. On l'évitera en utilisant des fichiers intermédiaires.

Etant donné que le temps calcul est peu sensible à une augmentation du nombre N_{ins} de contrôles et est proportionnel au nombre T de périodes, l'évaluation du gradient du critère par la méthode adjointe est beaucoup moins coûteuse que celle par différence finie (qui elle consomme un temps calcul proportionnel à $N_{ins} \cdot T^2$).

On remarquera cependant que la méthode proposée par Gabay et al nécessite :

- l'évaluation de toutes les matrices définies en (59)
- une analyse de structure pour pouvoir utiliser la méthode CPR pour cette évaluation
- le stockage de ces matrices
- la résolution des systèmes d'équations récurrents ou linéaires, mentionnés plus haut.

Chapitre IV

FORME QUASI-TRIANGULAIRE D'UN SYSTEME D'EQUATIONS

Remarque : Dans ce chapitre nous verrons entre autres l'importance de la correspondance biunivoque entre variables endogènes et équations construite au paragraphe 1 du chapitre III.

§ 0 RAPPELS DE LA THÉORIE DES GRAPHS

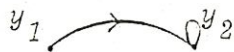
- On appelle *graphe orienté d'ordre n* un couple $G = (Y, U)$ constitué par :
 - un ensemble fini $Y = \{y_1, y_2, \dots, y_n\}$ dont les éléments sont appelés *sommets*
 - une famille finie $U = (u_1, u_2, \dots, u_m)$ d'éléments du produit cartésien, c'est-à-dire $u_i \in Y \times Y = \{(x, y) : x \in Y, y \in Y\}$

Les éléments u_i sont appelés *arcs*.

- On appelle *boucle* d'un graphe, un arc (y_i, y_i) .
- Nous pouvons représenter un graphe par un schéma :

Par exemple : $Y = \{y_1, y_2\}$

$$U = \{u_1 = (y_1, y_2), u_2 = (y_2, y_2)\}$$



L'arc u_1 est représenté par une flèche orientée de y_1 vers y_2 .

y_1 est l'extrémité initiale de l'arc u_1

y_2 est l'extrémité terminale de

l'arc u_1

- Un chemin $\mu = (u_1, u_2, \dots, u_i, u_{i+1}, \dots, u_q)$ est une suite d'arcs telle que l'extrémité terminale de u_i coïncide avec l'extrémité initiale de u_{i+1} .
 q est la longueur du chemin.
- Un circuit σ est un chemin $\sigma = (u_1, u_2, \dots, u_q) = [y_1, y_2, \dots, y_r]$ tel que l'extrémité terminale du dernier arc u_q coïncide avec l'extrémité initiale de premier arc u_1 .

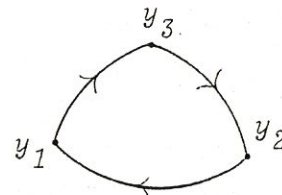
Exemple : $Y = \{y_1, y_2, y_3\}$

$$U = \{u_1, u_2, u_3\}$$

$$\text{où } u_1 = (y_1, y_2)$$

$$u_2 = (y_2, y_3)$$

$$u_3 = (y_3, y_1)$$



$\mu = (u_1, u_2)$ est un chemin de longueur 2.

$\sigma = (u_1, u_2, u_3) = [y_1, y_2, y_3, y_1]$ est un circuit.

- Un circuit est élémentaire lorsqu'il ne passe qu'une seule fois par chacun de ses sommets, sauf pour le dernier sommet qui coïncide avec le premier.
- Le sommet y_i est un prédécesseur du sommet y_j s'il existe un chemin reliant y_i à y_j .

Dans l'exemple ci-dessus, y_2 est un prédécesseur de y_1 et de y_3 .

Le sommet y_i est un successeur du sommet y_j s'il existe un chemin reliant y_j à y_i .

Dans l'exemple, y_2 est successeur de y_1 et y_3 .

Nous dirons que y_i est un prédécesseur ou successeur direct si le chemin reliant y_i à y_j ou y_j à y_i respectivement est de longueur 1.

Dans l'exemple, y_2 est prédécesseur direct de y_3 et successeur direct de y_1 .

- On appelle *demi-degré intérieur* du sommet y_i , noté d_i^- , le nombre d'arcs arrivant en y_i , c'est-à-dire dont y_i est une extrémité terminale.

On appelle *demi-degré extérieur* du sommet y_i , noté d_i^+ , le nombre d'arcs partant de y_i , c'est-à-dire dont y_i est extrémité initiale.

On appelle *degré* d'un sommet, la somme de son demi-degré intérieur et extérieur.

- On dit que y_i et y_j sont en relation de *connexité forte* ssi

un chemin reliant y_i à y_j et un chemin reliant y_j à y_i
ou ssi

un circuit σ contenant y_i et y_j

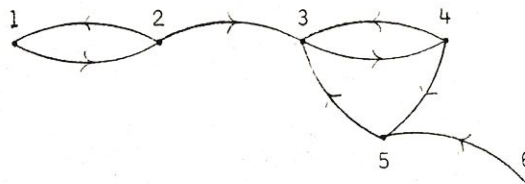
- Les classes d'équivalence de la relation de connexité forte sont appelées *composantes fortement connexes* de Y .
- On appelle *matrice associée au graphe* $G = (Y, U)$ la matrice booléenne

$$M = (m_{ij})$$

t.q.

$$m_{ij} = \begin{cases} 1 & \text{si } \exists \text{ un arc } u \in U \text{ t.q. } u = (y_j, y_i) \\ 0 & \text{sinon} \end{cases}$$

Exemple :



- y_2 est un prédécesseur de y_5 ; chemin : $[y_2, y_3, y_4, y_5]$

y_5 est un successeur direct de y_6 ; chemin : $[y_6, y_5]$

- Le demi-degré intérieur de y_3 , $d_3^- = 3$

- Il y a trois composantes fortement connexes :

$$C_1 = \{y_1, y_2\}$$

$$C_2 = \{y_3, y_4, y_5\}$$

$$C_3 = \{y_6\}$$

- La matrice M associée au graphe s'écrit :

$$\begin{array}{c} \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \end{array}$$

- On appelle *graphe condensé* $G'(Y', U')$ du graphe $G(Y, U)$, le graphe
 - dont les sommets sont les composantes fortement connexes de G
 - dont les arcs sont les liens entre les composantes fortement connexes

Plus précisément :

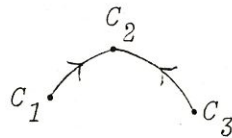
- $Y' = \{C_i \mid C_i \text{ est une composante fortement connexe de } G\}$
- \exists un arc $(C_i, C_j) \in U'$

ssi

$$\exists u \in U \text{ t.q. } u = (x_k, x_l) \text{ avec } x_k \in C_i \text{ et } x_l \in C_j$$

On peut démontrer que le graphe condensé est sans circuit.

Voici le graphe condensé de l'exemple précédent :



§ 0 DÉTERMINATION DU PROLOGUE ET DE L'ÉPILOGUE

Soit le modèle TOTO et sa matrice d'incidence E (cas initial I)

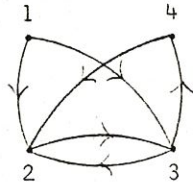
$$\left\{ \begin{array}{l} k_1 y_1 = f_1 \\ k_2 y_2 = f_2(y_1, y_3, y_4) \\ k_3 y_3 = f_3(y_1, y_2) \\ k_4 y_4 = f_4(y_3) \end{array} \right. \quad (17)$$

$$k_i = 1 \quad \forall i$$

$$E = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \end{array} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

A la matrice d'incidence E on peut faire correspondre un graphe à 4 sommets de façon à ce que E soit la matrice associée à ce graphe.

(60)



La relation entre le système d'équation (17) et le graphe (60) peut s'interpréter comme suit :

- Pour pouvoir tirer y_2 de l'équation 2, il faut connaître les valeurs des variables y_1, y_3, y_4 , ce qui correspond aux arcs arrivant en 2.
 - Comme la fonction f_1 ne fait pas intervenir de variables, on peut calculer immédiatement y_1 grâce à l'équation 1.
- D'ailleurs, il n'y a pas d'arcs arrivant en 1.

En général, nous pouvons donc dire que l'arc joignant l'élément i à l'élément j signifie qu'il faut connaître la valeur de y_i pour pouvoir tirer la valeur de y_j de son équation.

Cette interprétation explique le choix des éléments diagonaux :

Si $k_i = 0$, notre définition (12) entraîne $e_{ii} = 1$.

En effet, comme la variable y_i n'intervient pas dans l'équation i pour tirer sa valeur de cette équation, il faut connaître celle-ci déjà avant.

$e_{ii} = 1$ signifie qu'il y a une boucle dans le graphe.

Considérons le graphe condensé de (60) associé au système d'équations :



$$C_1 = \{1\}$$

$$C_2 = \{2, 3, 4\}$$

Puisque le graphe condensé est sans circuit, il y a au moins un sommet sans prédécesseurs, c'est-à-dire un sommet auquel n'arrive aucun arc. Il s'en suit que la résolution de ce bloc d'équations ne nécessite pas la connaissance des valeurs des variables des autres blocs. Une fois résolu ce bloc d'équations, on peut enlever le sommet correspondant du graphe condensé car les variables ne servent plus qu'en tant que constantes dans les autres équations. Le graphe réduit est toujours sans circuit et on peut recommencer le procédé.

La résolution d'un système de dimension n se ramène donc à la résolution successive dans un ordre bien déterminé des blocs d'équation correspondant aux composantes fortement connexes du graphe associé.

Pour TOTO, on résoudra d'abord l'équation 1 indépendamment des autres et puis le système formé par les équations 2, 3 et 4.

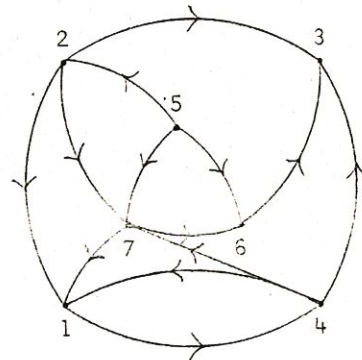
Pour les systèmes macro-économiques il y a généralement un gros bloc d'équations formant une composante fortement connexe, ainsi qu'un certain nombre de composantes fortement connexes constituées par une seule équation, dont une partie se trouve avant le gros bloc central dans le graphe condensé et l'autre après.

La première partie des équations "isolées" correspond à ce que nous avons appelé antérieurement prologue, le bloc central est constitué du coeur et des équations de bouclage et la partie des équations "isolées" suivant le bloc central est l'épilogue.

Voyons maintenant comment on peut détecter le prologue et l'épilogue.
Considérons l'exemple suivant :

$$E = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

Le graphe associé est le suivant :

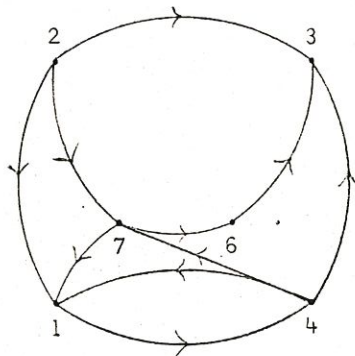


La première équation du prologue est celle dans laquelle aucune autre variable n'intervient. Dans le graphe le sommet correspondant à cette équation sera sans prédécesseurs (car on ne nécessite la connaissance d'aucune variable pour résoudre cette équation), c'est-à-dire que la ligne correspondante dans la matrice d'incidence est constituée d'éléments nuls. Nous voyons que le sommet 5 n'a pas de prédécesseur. Par définition de la matrice associée à un graphe, ceci implique que les éléments de la ligne 5 de E sont tous nuls. Nous allons permuer la matrice E pour amener la ligne et la colonne 5 en première place.

Nous obtenons :

$$E = \begin{matrix} & \begin{matrix} 5 & 1 & 2 & 3 & 4 & 6 & 7 \end{matrix} \\ \begin{matrix} 5 \\ 1 \\ 2 \\ 3 \\ 4 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

Considérons maintenant la matrice réduite et son graphe associé, c'est-à-dire le graphe initial dans lequel on supprime le sommet 5 ainsi que tous ses arcs, ou encore le graphe initial dans lequel on supprime la composante fortement connexe $\{5\}$.



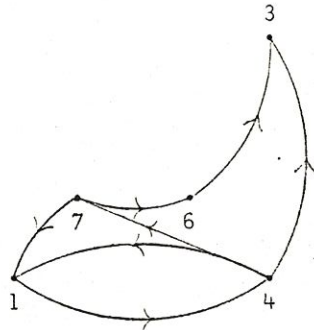
On constate que la ligne 2 de la matrice réduite est nulle. En effet, le sommet 2 n'a pas de prédécesseurs.

Nous allons donc permuter de façon à ce que la ligne et la colonne 2 se trouvent en deuxième place.

Nous obtenons :

$$E = \begin{matrix} & \begin{matrix} 5 & 2 & 1 & 3 & 4 & 6 & 7 \end{matrix} \\ \begin{matrix} 5 \\ 2 \\ 1 \\ 3 \\ 4 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

et le graphe réduit :



Dans la matrice réduite il n'y a plus de ligne nulle.

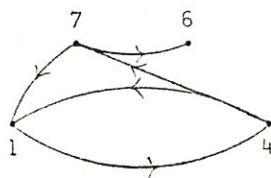
Comme il n'y a plus de sommet sans prédécesseurs, le prologue est donc déterminé. Il est constitué des équations 5 et 2.

La dernière équation de l'épilogue est celle dont la variable n'intervient dans aucune autre équation. Dans le graphe le sommet correspondant à cette équation sera sans successeurs (car on n'a jamais besoin de la valeur de cette variable pour en trouver une autre), c'est-à-dire que la colonne correspondante dans la matrice d'incidence est constituée de zéros.

C'est donc l'équation 3 qui devra figurer à la fin, car la colonne 3 est entièrement nulle. Après permutation, nous obtenons :

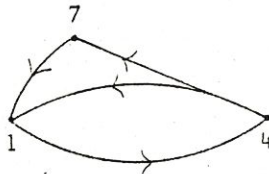
$$E = \begin{matrix} & 5 & 2 & 1 & 4 & 6 & 7 & 3 \\ \begin{matrix} 5 \\ 2 \\ 1 \\ 4 \\ 6 \\ 7 \\ 3 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

Le graphe réduit est le suivant :



On constate que tous les éléments de la colonne 6 de la matrice réduite sont nuls. D'ailleurs, c'est bien le sommet 6 qui n'a pas de successeurs. Nous allons donc permuter la matrice E pour amener la ligne et la colonne 6 en avant-dernière place. Nous obtenons la matrice et le graphe réduit :

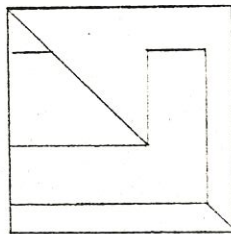
$$E = \begin{matrix} & \begin{matrix} 5 & 2 & 1 & 4 & 7 & 6 & 3 \end{matrix} \\ \begin{matrix} 5 \\ 2 \\ 1 \\ 4 \\ 7 \\ 6 \\ 3 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$



Le graphe réduit est fortement connexe. L'épilogue est donc constitué par les équations 6 et 3. En effet, il n'y a plus de colonne nulle dans la matrice d'incidence réduite.

Les équations 4 et 7 sont les équations de bouclage et l'équation 1 forme le coeur.

Remarquons que la matrice permutée a la forme décrite plus haut



et qu'elle contient deux variables de bouclage.

On notera qu'en mettant l'ordre 4,7,1 sur la matrice réduite, on obtient :

$$\begin{matrix} & \begin{matrix} 4 & 7 & 1 \end{matrix} \\ \begin{matrix} 4 \\ 7 \\ 1 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

c'est-à-dire une variable de bouclage.

Nous avons déjà remarqué au chapitre précédent qu'il est intéressant de réduire le nombre de variables de bouclage, même si le nombre minimum n'est pas nécessairement optimal.

Nous analyserons ce problème au paragraphe suivant.

§ 9 DÉTERMINATION DES VARIABLES DE BOUCLAGE

Après détermination du prologue et de l'épilogue, il nous reste une composante fortement connexe formant le coeur et les équations de bouclage.

Dans tout ce paragraphe, nous allons donc travailler avec une composante fortement connexe.

a) Caractérisation d'une variable de bouclage en termes de théorie des graphes

Rappelons notre définition de variable de bouclage :

- Intuitivement un ensemble de variables constitue un ensemble de variables de bouclage, si en fixant les valeurs de ces variables, le système devient récursif.
- Mathématiquement, y_j est une variable de bouclage

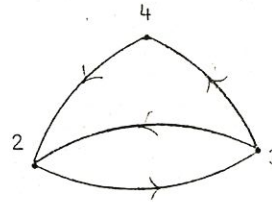
$$\text{ssi} \quad \prod_{i=1}^j (1 - e_{ij}) = 0 \quad (21)$$

où (e_{ij}) est la matrice d'incidence du système d'équations.

Reprenons l'exemple de TOTO (cas initial I).

Le prologue étant formé par l'équation 1 et l'épilogue étant vide, la matrice d'incidence réduite et le graphe réduit sont respectivement :

$$\begin{array}{c} 2 \ 3 \ 4 \\ \begin{array}{c} 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$



En adoptant l'ordre 4,2,3, la matrice d'incidence permutée s'écrit :

$$\begin{array}{c} 4 \ 2 \ 3 \\ \begin{array}{c} 4 \\ 2 \\ 3 \end{array} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

et ne contient plus qu'une variable de bouclage.

Le problème est d'avoir un algorithme qui trouve pour cet exemple un ensemble de variables de bouclage de taille 1.

Fixer une variable revient à enlever cette variable avec tous ses arcs du graphe, car il ne faut plus envisager que le système restant pour lequel cette variable n'est plus considérée comme variable, mais comme constante. On travaillera donc avec le graphe réduit.

Il est facile de voir que le système restant est récursif si le graphe réduit correspondant est sans circuit.

Pour trouver l'ordre de résolution récursive des équations, on procédera de la même manière que pour trouver l'ordre de résolution des blocs d'équations à partir du graphe condensé (voir paragraphe précédent).

En termes de théorie des graphes, nous pouvons donc caractériser un ensemble de variables de bouclage par le fait que le graphe réduit (on enlève toutes ces variables de bouclage ainsi que les arcs dont ils sont extrémités) est sans circuit.

Comme tout circuit contient nécessairement un circuit élémentaire, dans ce qui précède, on peut remplacer le mot "circuit" par "circuit élémentaire".

b) Procédé théorique

Une façon de procéder est de chercher tous les circuits élémentaires du graphe - soient A_i , pour $i \in I$, les sous-ensembles de l'ensemble des sommets Y définis par ces circuits - et un ensemble de taille minimale $A \subset Y$ tel que

$$A \cap A_i \neq \emptyset \quad \forall i \in I$$

Pour l'exemple TOTO, nous avons :

$$A_1 = \{2, 3\}$$

$$A_2 = \{2, 3, 4\}$$

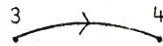
et on trouve

$$A = \{2\}$$

ou

$$A = \{3\}$$

Les graphes réduits sont respectivement :



Les matrices d'incidence dans l'ordre 3, 4, 2, respectivement 4, 2, 3 sont :

$$\begin{array}{c} 3 \ 4 \ 2 \\ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \end{array}$$

$$\begin{array}{c} 4 \ 2 \ 3 \\ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{array}$$

Ce procédé permet évidemment de trouver un ensemble de taille minimale, mais se révèle assez fastidieux dans le cas d'un graphe à plus de 3 sommets. En effet, déjà le nombre de circuits élémentaires est non polynomialement borné, et la recherche de l'ensemble A de taille minimale ne semble pas être évident non plus.

c) Algorithme de Ravelli

Nous expliquons ici le procédé proposé par Ravelli pour réduire le nombre de variables de bouclage.

On détermine d'abord un premier ensemble de variables de bouclage et on place les lignes et les colonnes correspondantes à la fin.

Puis on élimine les variables de bouclage superflues par le procédé qui suit, appliqué à toutes les variables de bouclage, en commençant par la première (celle la plus à gauche dans la matrice d'incidence) : deux cas se présentent :

1) L'élément diagonal de la variable testée est nul.

On détermine l'ensemble S des successeurs directs parmi les variables qui ne sont pas variables de bouclage et l'ensemble P de tous les prédécesseurs parmi les mêmes variables.

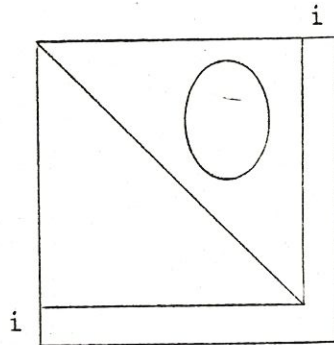
- Si $S \cap P$ est non vide, dans le graphe réduit aux variables du coeur et à la variable testée, il existe encore un circuit. La variable testée est donc une "vraie" variable de bouclage (car elle coupera un ou des circuits non coupés par les autres). Elle sera mise entre les variables de bouclage non testées et les autres "vraies" variables de bouclage.
- Par contre, si $S \cap P$ est vide, il n'y a plus de circuit dans ce graphe, c'est-à-dire que le système correspondant est récursif. La variable testée est alors une fausse variable de bouclage et est remise dans le coeur de façon à ce que celui-ci garde la forme triangulaire, c'est-à-dire que le coeur sera ordonné de manière à avoir dans l'ordre, les équations définissant P, l'équation correspondant à la variable testée, les équations définissant S.

2) L'élément diagonal de la variable testée est non nul.

La variable de bouclage est alors une vraie variable de bouclage et sera mise entre les variables de bouclage non testées et les autres "vraies" variables de bouclage.

Voyons maintenant comment déterminer S et P.

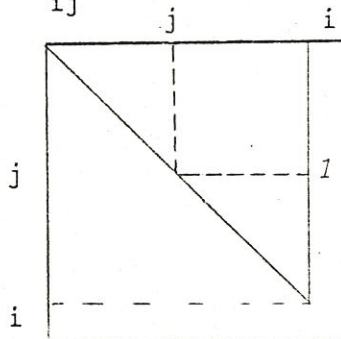
La matrice d'incidence réduite aux variables du coeur et à la variable qu'on teste a la forme suivante :



dans laquelle la colonne et la ligne i correspondent à la variable testée.

1. Les successeurs s'obtiennent en suivant les arcs partant de i dans le graphe réduit, ou encore en cherchant les éléments non nuls dans la colonne i de la matrice réduite, c'est-à-dire j est un successeur de i

$$\text{si } e_{ij} = 1.$$



2. Les prédécesseurs s'obtiennent en cherchant les extrémités initiales des arcs arrivant en i dans le graphe réduit, puis en cherchant les extrémités initiales de tous les arcs arrivant en un quelconque des sommets qu'on vient de déterminer, et ainsi de suite...

Rappelons que cette recherche se limite aux variables du coeur (la variable testée ne peut donc appartenir à P).

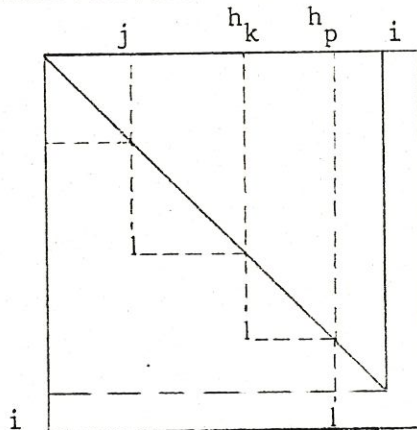
j est un prédécesseur de i

s'il existe une suite de sommets h_1, h_2, \dots, h_p

tels que

les éléments $e_{ih_1}, e_{h_1h_2}, \dots, e_{h_ph_j}$ valent 1 dans la matrice d'incidence réduite.

Remarquons que dans ce cas, les sommets h_1, h_2, \dots, h_p sont également des prédécesseurs de i .



Examinons maintenant concrètement la manière d'ordonner les éléments de P , l'équation i , les éléments de S dans le cas d'une fausse variable de bouclage.

Considérons l'exemple suivant pour lequel la matrice E est déjà réduite.

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	1
4	0	1	0	0	0	0	0	0	0	0
5	0	1	1	1	0	0	0	0	0	1
6	0	1	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	1	0	0	0	0
9	0	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	1	0

Nous testons la variable 10 (nouvelle numérotation).

Les variables de bouclage non testées portent les numéros 11, 12, ...

Par conséquent, 1 et 2 n'appartiennent pas au prologue mais au coeur (imaginons par exemple que les éléments (1, 11) et (2, 12) valent 1).

Pour la même raison, 5 et 8 n'appartiennent pas à l'épilogue.

Nous testons la variable 10.

$$S = \{3, 5\}$$

$$S \cap P = \emptyset$$

$$P = \{1, 7, 9\}$$

10 est une fausse variable de bouclage.

Il faudra avoir dans l'ordre 1, 7, 9, 10, 3, 5 en perturbant au minimum la matrice pour ne pas créer de nouvelles variables de bouclage.

(D'autres numéros s'intercaleront éventuellement entre les indices 1, 7, 9, 10, 3, 5).

Pour cela nous détectons le plus petit élément de l'ensemble des successeurs : s^* . Comme nous travaillons avec la nouvelle numérotation, les équations sont rangées dans l'ordre croissant.

Ici, c'est 3.

Nous amenons 10 juste avant 3 (permuter les lignes et les colonnes).

1 2 10 3 4 5 6 7 8 9

Nous cherchons alors parmi les prédécesseurs ceux qui sont plus grands que s^* .

Ici, 7 et 9 sont supérieurs à $s^* = 3$.

Nous amenons les lignes et colonnes 7 et 9 juste avant l'équation i, laquelle précède 3 immédiatement, en maintenant l'ordre croissant.

Ici nous ordonnons donc dans l'ordre :

1 2 7 9 10 3 4 5 6 8

Ceci donne la matrice d'incidence :

	1	2	7	9	10	3	4	5	6	8
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0
3	0	1	0	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0
5	0	1	0	0	1	1	1	0	0	0
6	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	1	0

Pour montrer que la manière d'ordonner, proposée ci-dessus, n'ajoute pas de variables de bouclage, il faut seulement examiner le cas où il y a eu perturbation de l'ordre croissant des équations.

Pour 10, il n'y a pas de problème : les successeurs 3 et 5 sont placés après et les prédécesseurs 1, 7, 9, avant par construction.

Nous avons détruit l'ordre croissant pour 3, 4, 5, 6 qui suivent 7, 9, 10.

Pour que 3 ne devienne pas variable de bouclage, il faut que les éléments $(7,3)$, $(9,3)$, $(10,3)$ ne valent pas un dans la matrice de départ. C'est le cas. Si $(7,3)$ valait un, 3 aurait été un prédécesseur de 10. De même pour $(9,3)$ et $(10,3)$. Le même raisonnement s'applique pour les variables 4, 5, 6.

Pour la variable 8, l'ordre croissant a été brisé, puisque 8 vient après 9 et 10.

De nouveau $(9,8)$ et $(10,8)$ ne pouvaient valoir un dans E, sinon 8 aurait été prédécesseur de 10.

Remarquons que si P est vide (les éléments à gauche de (i,i) dans la ligne i sont nuls), alors automatiquement i est une fausse variable de bouclage.

Nous plaçons la ligne i et la colonne i juste avant l'indice du premier successeur de i (l'élément $e_{ij} = 1$ avec j minimal (i fixé) fournit l'indice j en question).

L'algorithme de Ravelli peut s'écrire comme suit :

Etape 1

Déterminer E = matrice d'incidence du modèle

NL = nombre d'éléments non nuls sur chaque ligne (vecteur)

Initialiser $PERINI(I) = I$ pour chaque $I = 1, \dots, n$

Poser $l = 0$. l indique le nombre d'équations déjà placées dans le prologue.

Remarque : $PERINI$ est un vecteur qui conserve l'ancienne numérotation après les permutations. Ainsi $PERINI(I) = J$ signifie que la $i^{\text{ème}}$ équation de la nouvelle numérotation portait le numéro J au départ.

Etape 2 : Détermination du prologue

Si aucun des $NL(I)$ n'est nul pour $I = 1+l$ jusque n , aller à l'étape 3.

Sinon : a) Poser $l = 1+l$

b) Permuter les lignes et colonnes de E de la manière suivante :

La ligne I telle que $NL(I)$ est nul, est momentanément stockée dans un vecteur auxiliaire.

La ligne $I-1$ est amenée alors en I

La ligne $I-2$ est amenée en $I-1$

La ligne 1 est amenée en $1+l$

Le vecteur auxiliaire contenant la ligne I est recopié ensuite à la position 1 .

Nous effectuons la même opération sur les colonnes de E .

c) Permuter de la même manière le vecteur $PERINI$.

$PERINI(I)$ est stocké dans IA .

Puis $PERINI(I) = PERINI(I-1)$

\vdots

$PERINI(1+l) = PERINI(1)$

$PERINI(1) = IA$

d) Déterminer $NL(J)$ pour $J = 1+l, \dots, n$ où $NL(J)$ = nombre

d'éléments non nuls sur la ligne J , comptés entre les colonnes $1+l$ et n .

e) Retourner à l'étape 2.

N.B. : Si I (avec $NL(I) = 0$) était égal à 1, nous n'effectuerions pas les étapes 2b) ni 2c).

Etape 3

Si $1 = n$, aller à l'étape 9.

Le système est récursif. Nous résolvons les équations dans l'ordre $PERINI(1) \dots PERINI(n)$.

Sinon : le prologue contient les équations $PERINI(1) \dots PERINI(1)$ dans cet ordre

- poser $m = n$ (m indique le dernier numéro avant la première équation de l'épilogue)
- déterminer NC = nombre d'éléments non nuls sur chaque colonne (vecteur)
- aller à l'étape E

Etape 4 : Détermination de l'épilogue

Si aucun des $NC(I)$, pour I allant de $1+1$ à m n'est nul, aller à l'étape 5

Sinon, soit I tel que $NC(I) = 0$.

Si $I = m$, aller en c)

Sinon : a) Permuter les lignes et colonnes de E comme suit :

- copier la ligne I dans un vecteur auxiliaire,
- copier ensuite les lignes $I+1$ à la place de la ligne I , $(I+2)$ à la place de $(I+1)$, ..., m à la place $m-1$,
- puis copier le vecteur auxiliaire à la place de la ligne m .
- faire la même chose sur les colonnes.

b) Permuter de la même manière $PERINI$.

c) Déterminer $NC(J)$ pour $J = 1+1, \dots, m-1$ en ne tenant pas compte des éléments sur les lignes m à n .

d) Poser $m = m-1$

e) Aller à l'étape 4.

Etape 5 : Détermination d'un premier ensemble de variables de bouclage

Déterminer un premier ensemble de s variables de bouclage.

Pour cela :

- poser $s = 0$

- poser $il = m$

(1*)- pour chaque colonne i de il à $l+1$ (par pas de -1),

examiner s'il existe un j avec $l+1 \leq j \leq i$ tel que $E(i,j) = 1$

Si oui : • poser $s = s+1$ (nous avons une variable de bouclage en plus)

- si $i = m$, - poser $m = m-1$

- traiter la colonne suivante ($i-1$) :
retourner en (1*).

sinon : $i < il$

- mettre la colonne i de E (non réduite)
dans un vecteur auxiliaire ;

amener la colonne $i+1$ en position i ,

amener la colonne $i+2$ en position $i+1$,

amener la colonne m en position $m-1$,

copier le vecteur auxiliaire à la
place de la colonne m .

- permuter les lignes de E de la même
façon

- permuter de même le vecteur *PERINI*

- poser $m = m-1$

- aller en (1*) : traiter la colonne $i-1$

Sinon : aller en (1*) : traiter la colonne $i-1$

Etape 6 : Recherche des "vraies" variables de bouclage

Le nombre s d'équations de bouclage peut encore être réduit.

Nous allons tester chacune des variables de bouclage y_i , $i = m+1, \dots, m+s$.

m indique la dernière équation du coeur, i , la prochaine variable à

tester, il , la dernière position avant les variables de bouclage déjà

testées (épilogue, si nous commençons à tester).

Posons $i = m+1$.

Soit y_i la variable de bouclage correspondante que nous testons.

(Nous avons $il = m+s$ à la fin de l'étape 5).

1) Si $E(i,i) = 1$ (c'est-à-dire $k_i = 0$), y_i est une vraie variable de bouclage :

- Permuter les lignes et colonnes de la matrice E (non réduite) comme suit :
 - stocker la ligne i dans un vecteur auxiliaire,
 - amener la ligne $i+1$ en position i ,
 - amener la ligne $i+2$ en position $i+1$,
 - amener la ligne il en position $il-1$
 - copier le vecteur auxiliaire en position il
 - répéter ces permutations sur les colonnes.
- Permuter de manière similaire le vecteur *PERINI*
- Poser $il = il-1$.
- Si $il = m$, aller à l'étape 9.

Toutes les variables de bouclage ont été testées.

Sinon : retourner en 1), étape 6

(i n'est pas modifié : il indique toujours la prochaine variable à tester).

2) Sinon ($E(i,i) = 0$)

- Déterminer $P =$ l'ensemble des prédécesseurs de y_i
- Déterminer $S =$ l'ensemble des successeurs de y_i
- Si l'intersection de S et P est vide, aller à l'étape 7.
- Sinon, aller à l'étape 8.

Etape 7 : Traitement d'une fausse variable de bouclage

La variable y_i est une fausse variable de bouclage.

- Ordonner E de manière à avoir dans l'ordre, les équations définissant P , l'équation i , les équations définissant S .
- Permuter *PERINI* de la même manière.
- Poser $s = s-1$
- Poser $i=i+1$ (le nombre de variable de bouclage étant diminué de un, la position i de la prochaine variable à tester recule d'une place vers la droite).

Poser $m = m+1$ (le coeur possède une équation en plus).

Si $il = m$, aller à l'étape 9.

Toutes les variables de bouclage ont été testées.

Sinon, retourner à l'étape 6 en 1).

Etape_8 : Traitement d'une vraie variable de bouclage

La variable y_i est une vraie variable de bouclage.

Nous effectuons les mêmes opérations qu'à l'étape 6,1).

- Permuter les lignes et colonnes de E pour amener la variable y_i en position il .
- Permuter de la même façon *PERINI*.
- Poser $il = il-1$.
- Si $il = m$, aller à l'étape 9.

Toutes les variables ont été testées.

Sinon, aller à l'étape 6,1).

Etape_9

STOP

Nous allons illustrer les différentes étapes de l'algorithme à l'aide de l'exemple suivant.

A la figure ci-dessous, nous indiquons les différentes valeurs de NL et NC, au cours des étapes 1 à 4.

	1	2	3	4	5	6	7	NL	
$E =$	1	0	1	1	0	0	0	0	2
	2	0	0	0	0	0	0	1	1
	3	0	0	0	1	1	0	0	2
	4	0	0	1	0	1	0	0	2
	5	0	0	0	0	0	1	0	1
	6	0	0	0	0	0	0	1	1
	7	0	0	1	0	0	0	0	1
NC	0	1	3	1	2	1	2	$m=7$	
	0	2	1	2	1	2	$m=6$		
	1	1	2	1	2				

E ne possède pas de ligne nulle. Le prologue est donc vide (étape 2 et 3).

A l'étape 4, nous constatons que la première colonne de E est nulle ; d'où, l'équation 1 est amenée en position 7 (dans l'épilogue).

Nous déterminons l'épilogue qui contient les équations 2 et 1.

A l'étape 5, nous obtenons la matrice suivante :

$$\begin{array}{c}
 \begin{array}{c} 3 \ 4 \ 5 \ 6 \ 7 \ 2 \ 1 \\
 \begin{array}{l} 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 2 \\ 1 \end{array} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}
 \end{array}
 \end{array}
 \begin{array}{l}
 \text{système réduit} \\
 \text{épilogue}
 \end{array}$$

Les étapes 6 à 9 sont effectuées sur la matrice réduite :

$$\begin{array}{c}
 \begin{array}{c} 3 \ 4 \ 5 \ 6 \ 7 \\
 \begin{array}{l} 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}
 \end{array}
 \end{array}$$

Etape 6

- Déterminer les variables de bouclage sur la matrice réduite :
4, 5, 6, 7. Donc $s = 4$.
- Les équations correspondantes étant à la fin de la matrice réduite, il ne faut rien permuter..
- $m = 1$

Nous allons tester les variables de bouclage : 4, 5, 6, 7.

Etape 7

Pour tout i , $k_i = 1$.

D'où, k_i n'intervient pas dans la détermination des vraies variables de bouclage.

Variable 4

Prédécesseur : 3

Successeur : 3

La variable 4 est une vraie variable de bouclage.

La matrice réduite devient :

$$\begin{array}{c} 3 \ 5 \ 6 \ 7 \\ \begin{array}{l} 3 \left(\begin{array}{cccc} 0 & 1 & 0 & 0 \end{array} \right) \\ 5 \left(\begin{array}{cccc} 0 & 0 & 1 & 0 \end{array} \right) \\ 6 \left(\begin{array}{cccc} 0 & 0 & 0 & 1 \end{array} \right) \\ 7 \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right) \end{array} \end{array}$$

Variable 5

Prédécesseur : vide

Successeur : 3

La variable 5 est une fausse variable de bouclage..

Nous l'amenons dans le coeur en ordonnant les équations dans l'ordre

5 - 3 - 6 - 7

La matrice d'incidence devient :

$$\begin{array}{c} 5 \ 3 \ 6 \ 7 \\ \begin{array}{l} 5 \left(\begin{array}{cccc} 0 & 0 & 1 & 0 \end{array} \right) \\ 3 \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right) \\ 6 \left(\begin{array}{cccc} 0 & 0 & 0 & 1 \end{array} \right) \\ 7 \left(\begin{array}{cccc} 0 & 1 & 0 & 0 \end{array} \right) \end{array} \end{array}$$

Variable 6

Prédécesseur : vide

Successeur : 5

En ordonnant les équations, comme suit : 6 - 5 - 3 - 7 , nous amenons l'équation 6 dans le coeur.

La matrice réduite devient :

$$\begin{array}{c} 6 \ 5 \ 3 \ 7 \\ \begin{array}{l} 6 \left(\begin{array}{cccc} 0 & 0 & 0 & 1 \end{array} \right) \\ 5 \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \end{array} \right) \\ 3 \left(\begin{array}{cccc} 0 & 1 & 0 & 0 \end{array} \right) \\ 7 \left(\begin{array}{cccc} 0 & 0 & 1 & 0 \end{array} \right) \end{array} \end{array}$$

Variable 7

Prédécesseurs : 3 , 5 , 6

Successeur : 6

La variable 7 est une vraie variable de bouclage.

Toutes les variables de bouclage ont été testées.
STOP.

Nous avons déterminé :

- le prologue : vide
- le coeur : équations 6 , 5 , 3
- les équations de bouclage : 7 et 4
- l'épilogue : équations 1 et 2.

La matrice d'incidence du système global a la structure quasi-triangulaire :

	6	5	3	7	4	2	1
6				1	0		
5	1			0	0		
3	0	1		0	1		
7	0	0	1		0		
4	0	1	1	0			
2	0	0	0	1	0		
1	0	0	1	0	0	1	

REMARQUES IMPORTANTES :

=====

- Il est clair, qu'en pratique les permutations ne se feront jamais sur la matrice d'incidence. Toute l'information nécessaire est contenue dans le vecteur de permutation *PERINI*.
- L'accès à la matrice d'incidence E se fera par l'intermédiaire de la sous-routine ISE expliquée plus haut.

Notons que les équations supprimées (lors d'une exogénéisation) se trouveront dans le prologue après renumérotation (car la ligne correspondante est nulle).

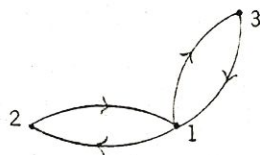
- Le calcul de NL et de NC ne se refait pas à chaque itération. En effet, un ajustement de NL et NC est moins coûteux et facile à réaliser. Ceci se fera respectivement dans les sous-routines LAJUST et CAJUST.
- La détermination des ensembles S et P se fera dans la sous-routine SUCPRE. Pour la recherche des prédécesseurs, on utilisera une méthode "Branch and Bound".
- Dans notre programme, à l'étape 9 se fera l'analyse de structure nécessaire pour l'évaluation du jacobien du système de bouclage. Cette étape sera expliquée dans le chapitre suivant.
- L'algorithme de Ravelli ne fournit pas toujours un ensemble de variables de bouclage de taille minimale.

La raison en est la suivante :

A l'étape 5 on détermine un premier ensemble de variables de bouclage. Les variables ne figurant pas dans cet ensemble ne pourront jamais devenir variables de bouclage.

Or, il est possible qu'une telle variable doive devenir variable de bouclage pour former un ensemble de taille minimale.

Nous explicitons cette possibilité dans l'exemple suivant :



$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{array}$$

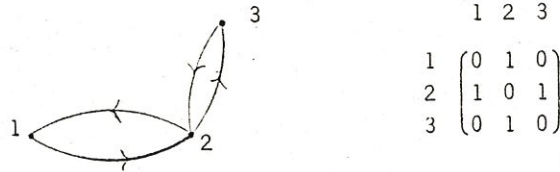
Le premier ensemble de variables de bouclage est formé par les éléments 2 et 3.

Il est facile de voir qu'elles sont toutes deux des vraies variables de bouclage.

L'algorithme fournira donc ces deux variables de bouclage.

D'autre part, il est évident que {1} est un ensemble de variables de bouclage de taille minimale.

Si la numérotation initiale était la suivante :



Le premier ensemble de variables de bouclage serait encore 2,3 .

2 est une vraie variable de bouclage et on permute donc E pour l'amener à la fin.

On obtient :

$$\begin{array}{c}
 \begin{matrix} & & 1 & 3 & 2 \\ 1 & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \\ 3 \\ 2 \end{matrix}
 \end{array}$$

3 est maintenant une fausse variable de bouclage.

Ici l'algorithme de Ravelli fournit donc bien un ensemble de variables de bouclage de taille minimale ({2}).

Cet exemple montre bien que le nombre de variables de bouclage fournies par l'algorithme de Ravelli dépend de l'ordre initial des équations.

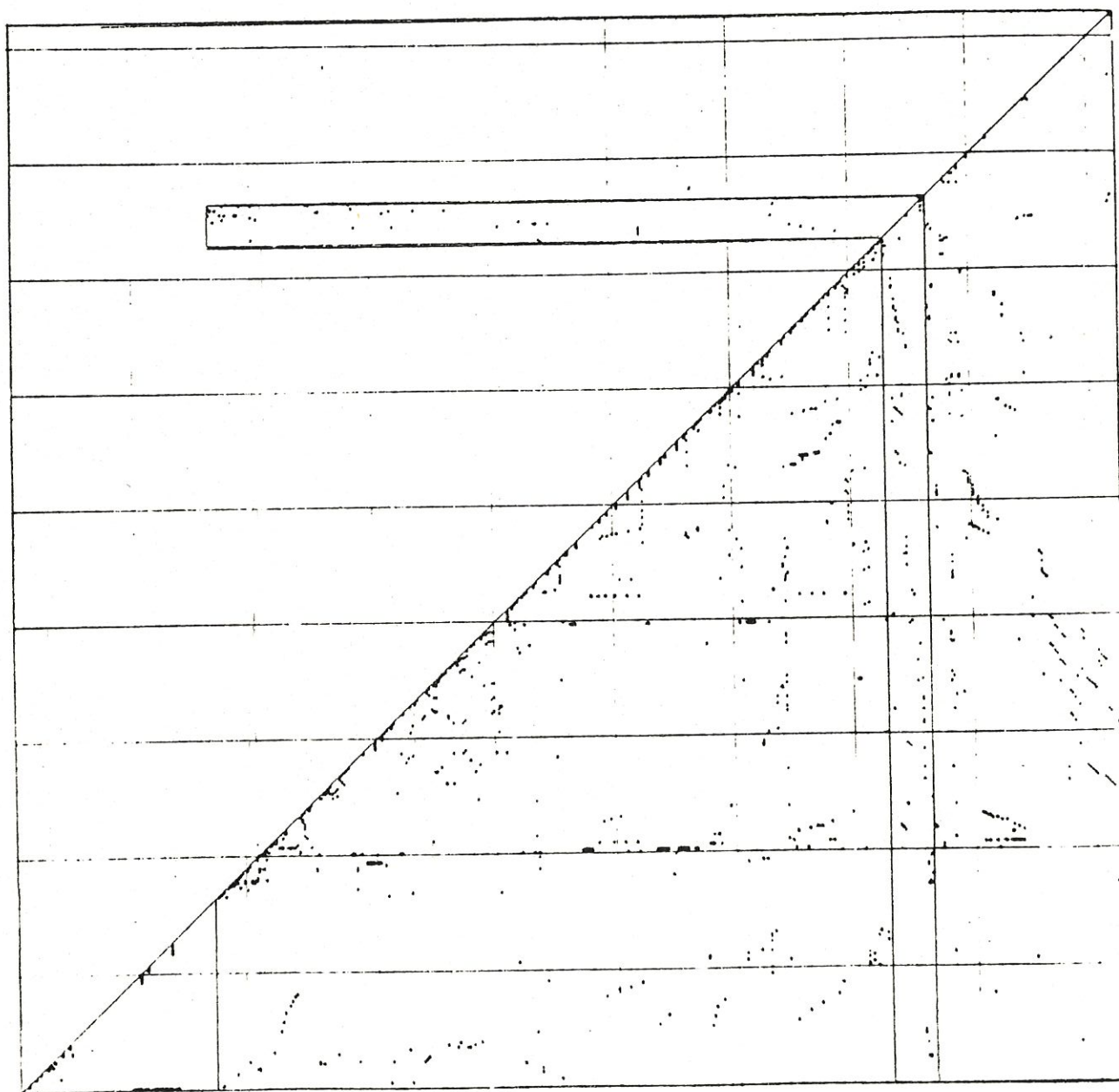
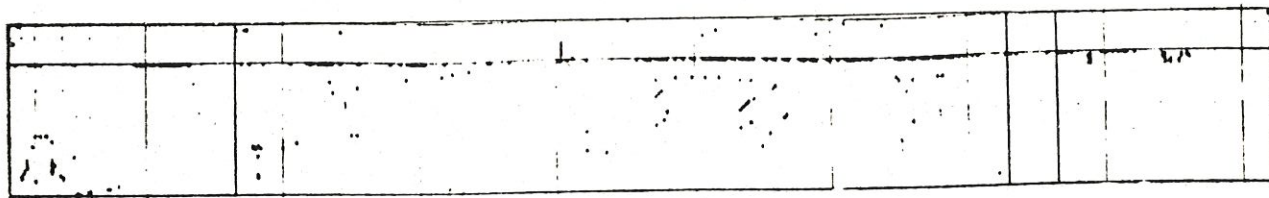
En effet, le premier ensemble de variables de bouclage (déterminé à l'étape 5) dépend forcément de l'ordre initial et il n'y a aucune raison pour que cet ensemble contienne un nombre minimum de vraies variables de bouclage.

C'est pour cette raison que nous permettons à l'utilisateur de fournir lui-même un ordre initial. Cette information sera contenue dans le vecteur *PERINI* (permutation initiale) qui ne sera alors pas initialisé à l'étape 1.

• Les résultats obtenus pour MARIBEL sont les suivants :

Le prologue comporte 82 équations et l'épilogue en compte 79.

Le coeur et les équations de bouclage sont formés par 300 équations.



- figure 7 -

- En prenant l'ordre alphabétique des variables comme ordre initial, l'algorithme donne 28 variables de bouclage.
- En prenant l'ordre donné par les équations (qui ne se trouvent pas dans l'ordre alphabétique dans la sous-routine MODEL), l'algorithme de Ravelli nous fournit 18 variables de bouclage.

La figure 7 reproduit la matrice d'incidence de MARIBEL après application de l'algorithme (18 variables de bouclage).

Ceci est le meilleur résultat obtenu jusqu'à présent.

En annexe, nous fournissons une copie de la sous-routine exécutant l'algorithme de Ravelli, appelée RAVELL.

d) Un autre algorithme

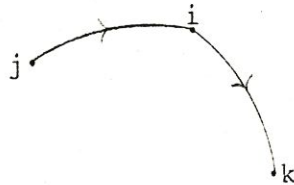
Soit un graphe fortement connexe.

L'idée du procédé est la suivante :

Suivant un critère que nous spécifierons plus loin, nous choisissons un sommet du graphe, soit le sommet i . Nous supprimons ce sommet en gardant l'information de tous les circuits auxquels il appartient, de la manière suivante :

les deux arcs (j,i) et (i,k) sont remplacés par un arc (j,k) , et cela pour tous les arcs ayant i respectivement comme extrémité terminale et initiale,

c'est-à-dire que le graphe



devient



Ainsi nous supprimons successivement des sommets *en gardant tous les circuits*. Comme nous éliminons des sommets à l'intérieur des circuits, ceux-ci vont nécessairement se réduire à des boucles.

Dès qu'un circuit se réduit à une boucle, nous choisissons le sommet correspondant comme variable de bouclage. Ensuite, nous éliminons ce sommet ainsi que tous les arcs dont il est une extrémité. Nous enlevons ainsi tous les circuits auxquels cette variable de bouclage appartenait.

Nous recommençons le procédé jusqu'au moment où tous les sommets sont éliminés.

Toute l'astuce réside dans le choix du prochain sommet à éliminer.

Comme le graphe de départ est fortement connexe, tous les sommets du graphe considéré ont toujours un demi-degré intérieur et extérieur plus grand ou égal à 1. Si ce n'était pas le cas pour un quelconque des sommets, ce sommet appartiendrait au prologue (si le demi-degré intérieur est zéro) ou à l'épilogue (si le demi-degré extérieur vaut zéro), et le graphe ne serait plus fortement connexe.

Ceci reste vrai tant qu'on n'enlève pas de variables de bouclage.

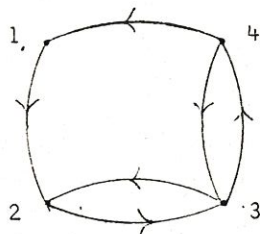
Dès qu'on élimine une variable de bouclage, le graphe réduit n'est plus nécessairement fortement connexe.

On commencera alors par l'élimination des sommets ayant un demi-degré intérieur ou extérieur égal à zéro (ces sommets ne se trouvent dans aucun circuit), et nous nous ramenons ainsi au cas où tous les sommets ont un demi-degré intérieur et extérieur plus grand ou égal à 1.

(Remarquons que même dans ce cas le graphe réduit n'est pas nécessairement fortement connexe).

Les demi-degrés intérieurs et extérieurs s'obtiennent en comptant le nombre d'éléments non nuls dans les lignes, respectivement dans les colonnes.

- Supposons que le sommet i ait un demi-degré extérieur égal à 1, c'est-à-dire qu'il n'ait qu'un successeur, soit k . Tous les circuits contenant le sommet i passent aussi par le sommet k . Il s'en suit que k est au moins une aussi "bonne" variable de bouclage que i . En enlevant le sommet i du graphe, nous ne perdons pas de sommets nécessaires pour former l'ensemble A de variables de bouclage de taille minimale.
- Un raisonnement analogue peut se faire pour un sommet ayant un demi-degré intérieur égal à 1 et son prédécesseur unique.
- Illustrons ceci à l'aide d'un exemple avant de continuer.
Soit le graphe et la matrice d'incidence associée :



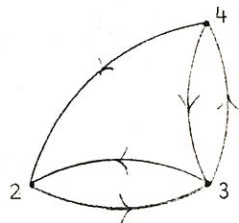
$$\begin{array}{cccc|c}
 & 1 & 2 & 3 & 4 & d_i^- \\
 1 & 0 & 0 & 0 & 1 & 1 \\
 2 & 1 & 0 & 1 & 0 & 2 \\
 3 & 0 & 1 & 0 & 1 & 2 \\
 4 & 0 & 0 & 1 & 0 & 1 \\
 d_i^+ & 1 & 1 & 2 & 2 &
 \end{array}$$

Les sommets 1, 2 et 4 sont candidats à l'élimination.

Dans le cas où il y a plusieurs candidats, nous choisissons le plus petit parmi ceux-ci.

Nous commençons donc le procédé par le sommet 1.

Après l'élimination du sommet 1, nous obtenons :



$$\begin{array}{ccc|c}
 & 2 & 3 & 4 & d_i^- \\
 2 & 0 & 1 & 1 & 2 \\
 3 & 1 & 0 & 1 & 2 \\
 4 & 0 & 1 & 0 & 1 \\
 d_i^+ & 1 & 2 & 2 &
 \end{array}$$

où l'arc $(4,2)$ remplace les arcs supprimés $(4,1)$ et $(1,2)$.

Pour obtenir la matrice d'incidence nous avons procédé comme suit :

Aux sommets successeurs de 1, nous rajoutons comme prédécesseurs les sommets prédécesseurs de 1.

Les successeurs de 1 s'obtiennent en cherchant les éléments non nuls dans la première colonne, dans notre cas 2 est le seul successeur. Les prédécesseurs de 2 sont les éléments non nuls dans la ligne 2, et ceux de 1 sont les éléments non nuls de la ligne 1.

Pour rajouter les prédécesseurs de 1 à ceux de 2, nous faisons l'opération "ou" sur les lignes correspondantes, c'est-à-dire

$$e_{12} \vee e_{22} \rightarrow e_{22}$$

$$0 \vee 0 \rightarrow 0$$

$$e_{13} \vee e_{23} \rightarrow e_{23}$$

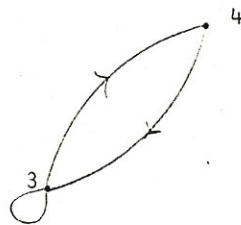
$$0 \vee 1 \rightarrow 1$$

$$e_{14} \vee e_{24} \rightarrow e_{24}$$

$$1 \vee 0 \rightarrow 1$$

Candidats à l'élimination : 2 et 4.

Nous éliminons le sommet 2. Nous obtenons :



$$\begin{array}{cc|c} & 3 & 4 & \bar{d}_i^- \\ \hline 3 & (1 & 1) & 2 \\ 4 & (1 & 0) & 1 \\ \hline \bar{d}_i^+ & 2 & 1 & \end{array}$$

où l'arc (3,3) remplace les arcs supprimés (2,3) et (3,2)

l'arc (4,3) déjà présent remplace les arcs (4,2) et (2,3).

En effet :

L'unique successeur de 2 est le sommet 3.

Donc

$$e_{23} \vee e_{33} \rightarrow e_{33}$$

$$1 \vee 0 \rightarrow 1$$

$$e_{24} \vee e_{34} \rightarrow e_{34}$$

$$1 \vee 1 \rightarrow 1$$

Le circuit $[2, 3, 2]$ du graphe initial s'est réduit à la boucle au sommet 3.

Nous choisissons donc 3 comme variable de bouclage et nous éliminons le sommet 3 du graphe, ainsi que tous les arcs dont il est une extrémité.

Nous obtenons le graphe réduit et la matrice réduite :

$$\begin{array}{c} 4 \\ 4 \quad (0) \end{array}$$

Nous supprimons le dernier sommet 4.

3 est donc l'unique variable de bouclage.

Pour obtenir l'ordre récursif du coeur, nous enlevons la colonne 3 et la ligne 3 de la matrice d'incidence et nous procédons comme pour la détermination du prologue.

Ceci nous fournit la matrice suivante :

$$\begin{array}{c} 4 \quad 1 \quad 2 \quad 3 \\ \begin{array}{l} 4 \\ 1 \\ 2 \\ 3 \end{array} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

- Supposons qu'à une étape donnée de notre procédé, il n'y ait plus de sommet avec un demi-degré intérieur ou extérieur plus petit ou égal à 1.

Le choix du prochain sommet à éliminer est alors basé sur l'idée suivante :

Si le degré du sommet k est inférieur à celui du sommet 1, le sommet 1 appartiendra probablement à plus de circuits que le sommet k . Le sommet 1 a donc plus de chances d'être une "bonne" variable de bouclage.

Nous éliminerons alors le sommet ayant un degré minimum.

- Après l'élimination de tous les sommets, nous avons déterminé l'ensemble des variables de bouclage.

Pour chercher l'ordre de récursivité du coeur, nous enlevons toutes les lignes et colonnes correspondant aux variables de bouclage de la matrice d'incidence et nous appliquons ensuite le procédé de détermination du prologue à la matrice réduite.

Nous pouvons maintenant donner l'algorithme :

Etape 1 - Etape 4

identiques à l'algorithme de Ravelli

Etape 5

Poser $IL = L + 1$

$IM = M$

IL et IM déterminent les bornes de la matrice réduite sur laquelle on travaille.

Initialiser NC et NL pour la matrice réduite.

Aller à l'étape 6.

Etape 6 : Détermination des variables de bouclage

Poser $I = IL$

a) Si $I > IM$, aller à l'étape 7

Sinon : si $E(I, I) = 1$, aller à b)

sinon : poser $I = I + 1$

aller à a)

b) I est une variable de bouclage

Ajuster *PERINI* pour amener I à la place de IM.

Permuter E, NL et NC de la même manière.

Poser $IM = IM - 1$.

Si $IL > IM$, aller à l'étape 11.

Sinon : ajuster NL et NC pour la matrice réduite

aller à a)

Etape 7 : Recherche d'un élément I tel que d_i^+ ou d_i^- vaut 0

Poser $I = IL$

a) Si $I > IM$, aller à l'étape 8.

Sinon : si $NL(I) = 0$ ou $NC(I) = 0$, aller à l'étape 10

sinon : $I = I + 1$
aller à a)

Etape 8 : Recherche d'un élément I tel que d_i^+ ou d_i^- vaut 1

Poser $I = IL$

a) Si $I > IM$, aller à l'étape 8.

Sinon : si $NL(I) = 1$ ou $NC(I) = 1$, aller à l'étape 10

sinon : $I = I + 1$
aller à a)

Etape 9 : Recherche d'un élément I tel que d_i soit minimal

Poser $MIN = \infty$

Poser $I = IL$

a) Si $I > IL$, aller à b)

Sinon : si $(NL(I) + NC(I)) < MIN$, poser $MIN = NL(I) + NC(I)$

poser $IMIN = I$

poser $I = I + 1$

aller à a)

sinon : poser $I = I + 1$
aller à a)

b) Poser $I = IMIN$.

Aller à l'étape 10.

Etape 10 : Elimination du sommet I

Ajuster *PERINI* pour amener I à la place de IL .

Permuter E , NL et NC de la même manière.

Poser $IL = IL + 1$.

Si $IL > IM$, aller à l'étape 11.

Sinon : poser $I = IL$

a) Si $I > IM$, aller à e)

Sinon : si $E(I, IL-1) = 1$, aller à b)

sinon : poser $I = I + 1$

aller à a)

b) Poser $J = IL$

c) Si $J > IM$, aller à d)

Sinon : poser $E(J, I) = E(J, IL-1) \vee E(J, I)$

poser $J = J + 1$

aller à c)

d) Ajuster $NL(I)$

Poser $I = I + 1$

Aller à a)

e) Ajuster NC pour la matrice réduite

Aller à l'étape 6

Etape 11

Etablir l'ordre récursif dans la matrice réduite de $L + 1$ à IM

(analogue à l'étape 1 et 2).

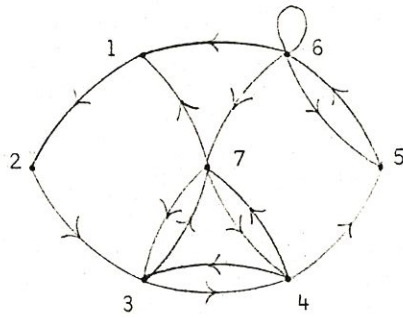
Aller à l'étape 12.

Etape 12

STOP

Nous allons illustrer les différentes étapes (5 à 11) de l'algorithme dans l'exemple qui suit.

Nous partons d'un graphe fortement connexe :



	1	2	3	4	5	6	7	NL
1	0	0	0	0	0	1	1	2
2	1	0	0	0	0	0	0	1
3	0	1	0	1	0	0	1	3
4	0	0	1	0	0	0	1	2
5	0	0	0	1	0	1	0	2
6	0	0	0	0	1	1	0	2
7	0	0	1	1	0	1	0	3

NC 1 1 2 3 1 4 3
↑

Etape 5

L'initialisation de NC et NL est faite dans le vecteur ligne et le vecteur colonne à côté de la matrice d'incidence.

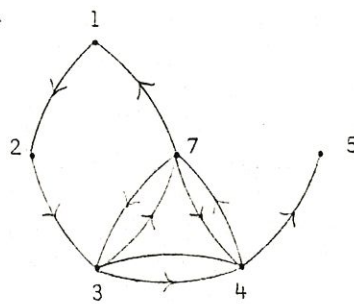
Etape 6

Le sommet 6 a un 1 sur la diagonale. C'est donc une variable de bouclage.

Nous permutons E de façon à placer la colonne et la ligne 6 à la fin.

Nous effectuons les mêmes permutations sur NL et NC.

Les composantes du vecteur PERINI peuvent se lire sur la matrice d'incidence.



	1	2	3	4	5	7	6	NL
1	0	0	0	0	0	1		1
2	1	0	0	0	0	0		1
3	0	1	0	1	0	1		3
4	0	0	1	0	0	1		2
5	0	0	0	1	0	0		1
7	0	0	1	1	0	0		2
6							1	

NC 1 1 2 3 0 3
↑

Etape 7

Le sommet 5 a un demi-degré extérieur égal à 0.

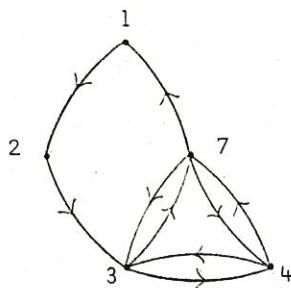
En effet, $NC(5) = 0$.

Aller à l'étape 10

Etape 10

On élimine le sommet 5 en le plaçant au début.

Nous obtenons le graphe réduit et la matrice d'incidence réduite.



	5	1	2	3	4	7	6	NL
5								
1	0	0	0	0	1			1 ←
2	1	0	0	0	0			1
3	0	1	0	1	1			3
4	0	0	1	0	1			2
7	0	0	1	1	0			2
6								
NC	1	1	2	2	3			

On va à l'étape 8.

Etape 8

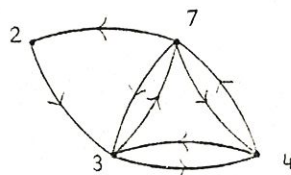
Il y a deux candidats à l'élimination ayant un demi-degré intérieur et extérieur égal à 1.

Choisissons le sommet 1 et allons à l'étape 10.

Etape 10

Nous éliminons le sommet 1.

Nous obtenons :



	5	1	2	3	4	7	6	NL
5								
1								
2				0	0	0	1	1 ←
3				1	0	1	1	3
4				0	1	0	1	2
7				0	1	1	0	2
6								
NC			1	2	2	3		

où l'arc (7,2) remplace les arcs (7,1) et (1,2).

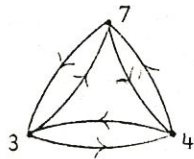
Dans la matrice d'incidence réduite, nous avons fait l'opération "ou" sur les lignes 1 et 2.

Etape 8

Le sommet 2 a un demi-degré extérieur valant 1.
Aller à l'étape 10.

Etape 10

Nous éliminons le sommet 2.
On obtient :



	5	1	2	3	4	7	6	NL
5								
1								
2								
3				0	1	1		2
4				1	0	1		2
7				1	1	0		2
6								
NC				2	2	2		

où les arcs $(7,2)$ et $(2,3)$ sont remplacés par l'arc $(7,3)$ déjà présent.

La matrice réduite a été obtenu en faisant :

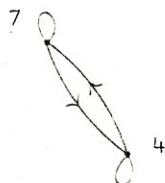
ligne 7 \vee ligne 2 \rightarrow ligne 7

Etape 9

Les 3 sommets ont un degré 4.
Nous choisissons le sommet 3.

Etape 10

Après élimination du sommet 3, nous obtenons :



	5	1	2	3	4	7	6	NL
5								
1								
2								
3								
4					1	1		2
7					1	1		2
6								
NC					2	2		

Etape 6

Le sommet 4 est une variable de bouclage.

Nous permutons donc E de manière à placer 4 avant 6,
donc

Q_7

$$\begin{array}{cccccccc}
 & 5 & 1 & 2 & 3 & 7 & 4 & 6 & NL \\
 \begin{array}{c} 5 \\ 1 \\ 2 \\ 3 \\ 7 \\ 4 \\ 6 \end{array} & \left(\begin{array}{ccccccc} & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & 1 & & \\ & & & & & & \\ & & & & & & \end{array} \right) & \begin{array}{c} \\ \\ \\ \\ 1 \\ \\ \end{array} \\
 NC & & & & & & 1 & &
 \end{array}$$

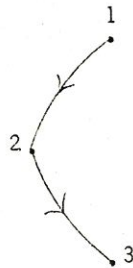
Le sommet 7 est variable de bouclage.

Tous les sommets du graphe sont maintenant éliminés.

Nous allons à l'étape 11.

Etape 11

On travaille sur le graphe réduit et la matrice réduite au coeur :



. 5

$$\begin{array}{cccccccc}
 & 5 & 1 & 2 & 3 & 7 & 4 & 6 \\
 \begin{array}{c} 5 \\ 1 \\ 2 \\ 3 \\ 7 \\ 4 \\ 6 \end{array} & \left(\begin{array}{ccccccc} 0 & 0 & 0 & 0 & & & \\ 0 & 0 & 0 & 0 & & & \\ 0 & 1 & 0 & 0 & & & \\ 0 & 0 & 1 & 0 & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{array} \right)
 \end{array}$$

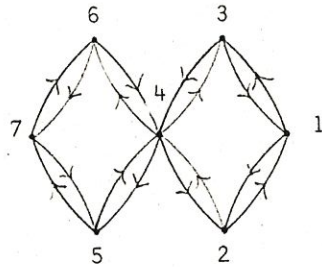
pour déterminer l'ordre récursif du coeur.

En fait, la matrice d'incidence est déjà sous la bonne forme :

$$\begin{array}{cccccccc}
 & 5 & 1 & 2 & 3 & 7 & 4 & 6 \\
 \begin{array}{c} 5 \\ 1 \\ 2 \\ 3 \\ 7 \\ 4 \\ 6 \end{array} & \left(\begin{array}{ccccccc} 0 & & & & 1 & 1 \\ & 0 & & & 1 & 1 \\ & 1 & 0 & & & & \\ & & 1 & 0 & 1 & 1 \\ & & & 1 & 0 & 1 & 1 \\ & & & 1 & 1 & 0 & \\ 1 & & & & & & 1 \end{array} \right)
 \end{array}$$

Comme l'algorithme de Ravelli, cet algorithme ne trouve pas un ensemble optimal de variables de bouclage.

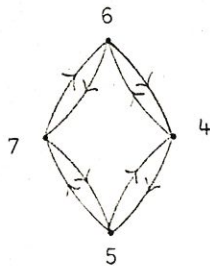
En voici un contre-exemple :



Il est facile de voir que l'ensemble $\{1, 4, 7\}$ forme un ensemble de variables de bouclage de taille minimale.

Or, l'algorithme à l'étape 9 va éliminer le sommet 1 du graphe..

Les sommets 2 et 3 deviennent alors variables de bouclage et le graphe réduit a la forme :



Pour rendre ce graphe récursif, il faut au moins deux variables de bouclage.

L'algorithme proposé va donc fournir 4 variables de bouclage.

Remarquons que si on avait éliminé 2 au lieu de 1 à l'étape 9 (2 était aussi bien candidat à l'élimination que 1), on aurait trouvé le bon ensemble.

Cependant nous avons :

Théorème

Soit le graphe simple d'ordre n fortement connexe, $G = (V, U)$, et la matrice associée E .

S'il est possible de rendre E quasi-triangulaire avec une seule variable de bouclage,

alors l'algorithme fournira un ensemble de variables de bouclage de taille 1.

Lemme

Soit un graphe $G = (V, U)$ d'ordre n ($n \neq 0$) fortement connexe, sans boucle tel que

$$\forall i = 1, \dots, n : d_i^+ \geq 2 \text{ et } d_i^- \geq 2$$

Alors pour rendre la matrice associée E quasi-triangulaire, il faut au moins deux variables de bouclage.

Démonstration du lemme

- a) Selon les hypothèses du théorème, on a : $n \geq 3$ (évident).
- b) Comme le graphe est fortement connexe, tout sommet appartient à au moins un circuit.
- c) Prenons un sommet quelconque i et montrons qu'il existe au moins un circuit qui ne passe pas par i .

Pour ceci, choisissons un sommet k_1 . Par hypothèse, il existe au moins deux arcs ayant k_1 comme extrémité initiale. Un de ces deux arcs a une extrémité finale k_2 différente de i . De la même manière, nous déterminons k_3, \dots, k_j, \dots . Comme il y a un nombre fini de sommets, en continuant la construction de la suite de ces sommets, nous repassons nécessairement par un des sommets k_j déjà trouvés. Ainsi, nous avons construit un circuit. Par construction, ce circuit ne passe pas par i .

i ne peut donc être l'unique variable de bouclage.

Comme le sommet i était quelconque, l'ensemble des variables de bouclage ne peut donc être de taille 1. ■

Démonstration du théorème

1^{ier} cas : Il existe i tel que $e_{ii} = 1$.

L'étape 6 de l'algorithme va détecter le sommet i , qui correspond par hypothèse à l'unique variable de bouclage.

2^{ième} cas : $e_{ii} = 0 \quad \forall i = 1, \dots, n$

Le lemme montre que le choix des sommets à éliminer se fait nécessairement à l'étape 7 ou 8 (jamais à l'étape 9) tant qu'on n'élimine pas de variables de bouclage.

Or, ni l'étape 7, ni l'étape 8 ne supprime des variables nécessaires pour former un ensemble de taille minimale.

La première variable de bouclage détectée est donc l'unique, ce qui démontre le théorème. ■

REMARQUES IMPORTANTES :

- Ici, les permutations ne se feront pas non plus vraiment sur la matrice d'incidence, mais les opérations "ou" doivent être effectuées réellement sur la matrice E .
Ceci est un désavantage par rapport à l'algorithme de Ravelli.
- Il semble que ce procédé donne moins de variables de bouclage que l'algorithme de Ravelli.
Jean-Marc Paul au Bureau du Plan a testé l'algorithme sur MARIBEL et a obtenu 16 variables de bouclage.
- Un gain de deux équations de bouclage n'est pas énorme, mais l'algorithme présente un autre avantage. Nous avons déjà mentionné qu'un nombre minimum de variables de bouclage n'est pas nécessairement optimal. Cependant, il serait peut-être intéressant, d'avoir des variables économiquement importantes comme variables de bouclage, ce qui pourrait accélérer la convergence de l'algorithme de résolution.

Pour réaliser cela, on pourrait donner un poids économique à chaque variable pour influencer le choix des sommets à éliminer :

- Au lieu de choisir à l'étape 8 un sommet au hasard entre les candidats à l'élimination (c'est-à-dire entre les sommets tels que $d_i^+ = 1$ ou $d_i^- = 1$), on pourrait choisir le candidat ayant un poids économique minimal.
- De même, ce serait le poids économique du sommet qui déciderait de l'élimination à l'étape 9 (en choisissant entre les sommets ayant un degré minimal).
- Une alternative à cette dernière possibilité est d'inverser les rôles du degré et du poids économique, c'est-à-dire de choisir le sommet de degré minimal parmi les sommets de poids minimal.

Nous n'avons malheureusement pas eu l'occasion de tester toutes ces possibilités.

Chapitre V

RESOLUTION DU MODELE

§ 0 ACCÈS AUX ÉQUATIONS

- a) Nous avons déjà mentionné que les équations se trouvent dans une sous-routine appelée MODEL.

Nous savons également que vu la structure quasi-triangulaire du modèle, il faut toujours évaluer des blocs d'équations, par exemple, pour résoudre le prologue, il faut évaluer les équations de 1 à 1.

Pour cette raison, l'accès à la sous-routine MODEL se fait avec deux paramètres : N1 et N2. On résoudra les équations de N1 à N2.

D'autre part, par exemple pour la résolution du prologue les numéros 1 et 1 correspondent à la nouvelle numérotation (après application de l'algorithme de renumérotation). Or nous n'avons bien sûr pas changé l'ordre des équations dans la sous-routine MODEL (qui serait d'ailleurs différent pour chaque problème particulier). Le lien entre l'ancien et le nouvel ordre est donné par le vecteur *PERINI*. Dès lors, on résoudra les équations de *PERINI*(N1) à *PERINI*(N2).

- b) Lors de l'explication du problème d'exogénéisation nous avons proposé de ne pas supprimer réellement les équations non utilisées dans la sous-routine MODEL, car sinon il faudrait changer celle-ci pour chaque problème particulier. D'ailleurs, nous nous sommes arrangé pour que l'algorithme de renumérotation mette les équations non utilisées dans le prologue. Cependant, lors de la résolution du prologue, il ne faudra pas les évaluer.

Nous avons également vu qu'une équation ne se résout pas nécessairement par rapport à sa variable d'assignation, mais éventuellement par rapport à une autre variable qui lui correspond.

Toute cette information sera contenue dans le vecteur *IFEQUA*, défini ci-dessous :

$$(61) \quad IFEQUA(I) = \begin{cases} 0 & \text{si l'équation } I \text{ est supprimée} \\ -1 & \text{si'il suffit de faire l'assignation} \\ J & \text{si'il faut résoudre l'équation } I \text{ par rapport} \\ & \text{à la variable } J \text{ qui n'est pas sa variable} \\ & \text{d'assignation} \end{cases}$$

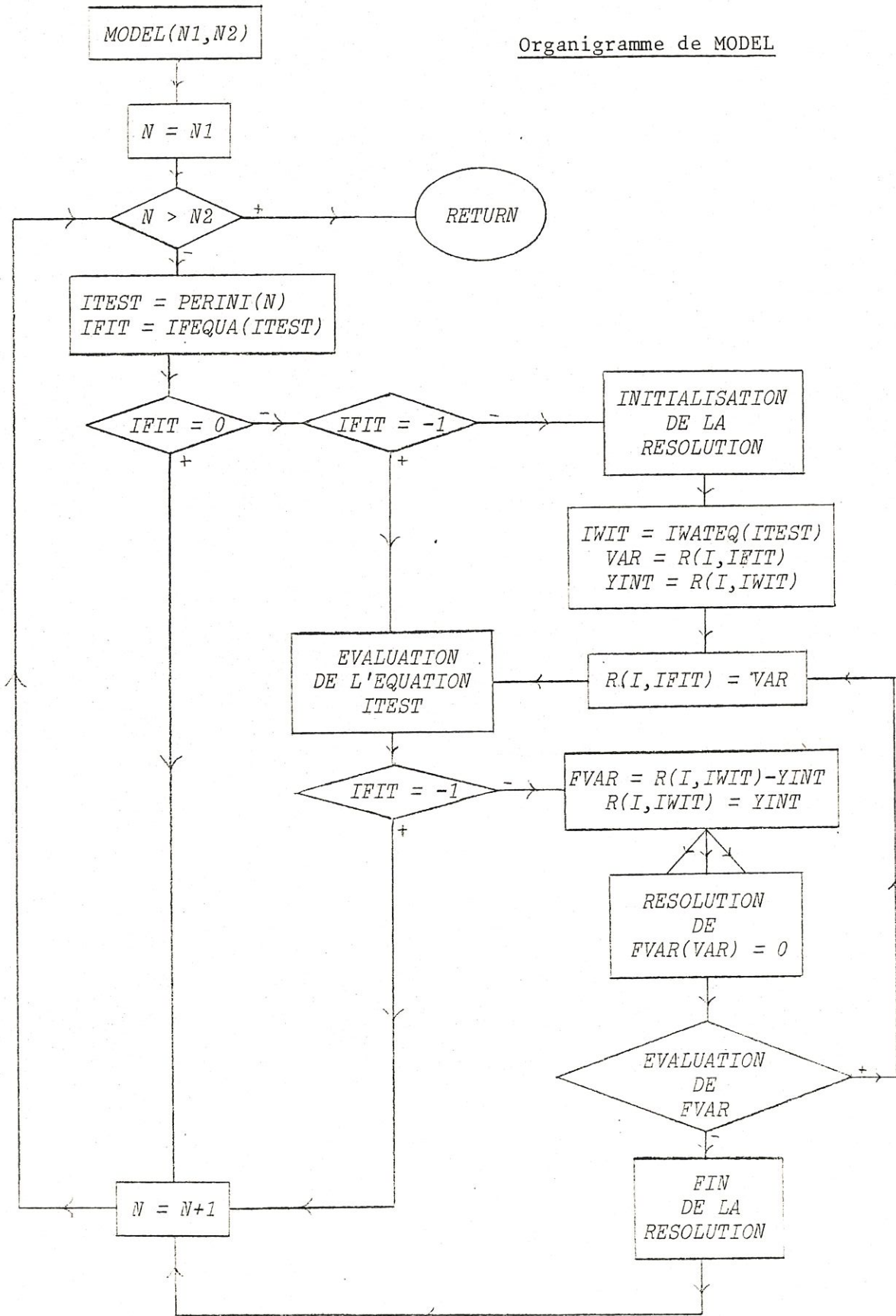
La sous-routine MODEL résout successivement les équations de N1 à N2. Avant de traiter une équation particulière parmi celles-ci, nous évaluons la composante correspondante du vecteur *IFEQUA* et nous procédons différemment suivant la valeur de celle-ci.

L'organigramme de la sous-routine MODEL est représenté par la figure 8.

Commentaires :

- ITEST est le numéro de l'équation qu'on traite.
- IWIT est le numéro de la variable d'assignation de l'équation traitée ITEST.
- Les cas IFIT = 0 et IFIT = -1 sont évidents.
- Cas d'une résolution
IFIT = numéro de la variable par rapport à laquelle il faut résoudre l'équation ITEST.
On a nécessairement : IFIT \neq IWIT
R(I,J) contient la valeur de la variable J à la période I.
Pour résoudre l'équation, nous itérons sur la valeur de VAR = R(I,IFIT).
Dans FVAR nous stockons la valeur de la fonction à annuler au point VAR, obtenue en faisant la différence entre R(I,IWIT) (valeur obtenue après assignation de l'équation) et YINT (valeur initiale de R(I,IWIT), c'est-à-dire avant l'assignation).
- La case "résolution de FVAR(VAR) = 0" sera expliquée au chapitre suivant.

Organigramme de MODEL



c) Comment construire le vecteur *IFEQUA* ?

Le cas où $IFEQUA(I) = 0$ est évident (cette information est donnée par $IRAV(I) = 0$)

1^{ier} cas : I n'est pas une équation de bouclage

- On a nécessairement que $IRAV(I) > 0$
- Si $IRAV(I) = IWATEQ(I)$,
alors la variable correspondant à l'équation I est sa variable d'assignation,
c'est-à-dire $IFEQUA(I) = -1$.
- Sinon il faut résoudre l'équation I par rapport à la variable $IRAV(I)$ (qui n'est pas sa variable d'assignation),
c'est-à-dire $IFEQUA(I) = IRAV(I)$.

2^{ième} cas : I est une équation de bouclage

Pour évaluer les fonctions $E_i(yb)$ pour des valeurs de yb fixées, il faut :

- résoudre le coeur
- évaluer les équations de bouclage

$$fb_i - l_i yb_i$$

Rappelons que ceci est la forme théorique de l'équation de bouclage.

En réalité, dans MODEL nous avons :

$$y_{IWATEQ(I)} = g_i$$

$$+ \text{ Si } l_i = 0$$

$$\begin{aligned} & fb_i - l_i yb_i \\ & = fb_i \\ & = g_i - y_{IWATEQ(I)} \end{aligned}$$

Pour évaluer les équations de bouclage nous procédons comme suit :

- poser $y_{int} = y_{IWATEQ(I)}$
- appel à $MODEL(I,I)$ qui doit faire l'assignation de l'équation I
- calculer $y_{IWATEQ(I)} - y_{int}$
- poser $y_{IWATEQ(I)} = y_{int}$

Dans ce cas-ci, il faut donc : $IFEQUA(I) = -1$.

+ Si $l_i = 1$ et $IRAV(I) = IWATEQ(I)$

i.e. $y_{b_i} = y_{IWATEQ(I)}$

et

$fb_i = g_i$

Nous procédons de la même manière que dans le cas $l_i = 0$.

Nous posons alors : $IFEQUA(I) = -1$.

+ Si $l_i = 1$ et $IRAV(I) \neq IWATEQ(I)$

Pour évaluer $fb_i - y_{b_i}$, il faut stocker la valeur de y_{b_i} dans une variable intermédiaire yint, tirer la valeur y_{b_i} de l'équation $y_{IWATEQ(I)} = g_i$,

faire la différence entre y_{b_i} et yint, et ensuite remettre la valeur de yint dans y_{b_i} .

Mais le système réduit des équations de bouclage

$$E_i(y_{b_i}) = 0$$

est résolu

si les valeurs données aux variables de bouclage sont telles que celles-ci vérifient les équations de bouclage après passage dans le coeur, c'est-à-dire que :

$$fb_i - y_{b_i} = 0$$

ou encore

$$g_i = y_{IWATEQ(I)}$$

DONC :

annuler la fonction $(fb_i - yb_i)$ définie implicitement par l'intermédiaire du coeur

est équivalent à

annuler la fonction $(g_i - y_{IWATEQ(I)})$ définie implicitement à travers le coeur.

Pour épargner du temps calcul, nous pouvons donc travailler avec la fonction $g_i = y_{IWATEQ(I)}$.

Nous posons donc $IFEQUA(I) = -1$ et nous procédons comme au cas précédent pour évaluer la fonction.

Conclusion : Si I est une équation de bouclage, nous posons toujours $IFEQUA(I) = -1$.

Illustrons la construction du vecteur IFEQUA à l'aide du modèle TOTO.

- Pour le cas initial (I), il est facile de voir que le vecteur IFEQUA vaut identiquement -1 (car $IWATEQ \equiv IRAV$).
- Envisageons maintenant le cas II 1).
L'équation 3 ainsi que sa variable sont supprimées.
 $IFEQUA(3)$ vaut donc 0.
- Reprenons le cas II 2)a).
L'équation 3 devrait se résoudre par rapport à une variable, y_3 , qui n'est pas la variable d'assignation.
Mais y_3 ne figure pas dans l'équation 3 et sera dès lors automatiquement une variable de bouclage.
 $IFEQUA(3)$ devra donc prendre la valeur -1.

- Pour clôturer, reprenons encore le cas II 2)b).

L'équation 3 se résoud par rapport à la variable y_6 qui y figure, mais qui n'est pas sa variable d'assignation.

La valeur de IFEQUA(3) dépendra de l'ordre des équations après la renumérotation :

- si l'équation 3 devient équation de bouclage, IFEQUA(3) vaudra -1
- sinon, IFEQUA(3) prendra la valeur 6.

REMARQUE IMPORTANTE :

=====

Dans l'équation adjointe (34), la matrice K ne représente rien d'autre

que $\left(\frac{\partial g_i}{\partial y_i} \right)_i$

si on travaille avec la forme théorique du modèle :

$$k_i y_i = f_i \quad \text{ou} \quad g_i = f_i - k_i y_i = 0$$

Les dérivées partielles $\frac{\partial g_i}{\partial y_i}$ valent en effet k_i , soit 0 ou 1.

Si l'on travaille avec les équations de bouclage telles qu'elles se trouvent dans la sous-routine MODEL, ceci n'est plus nécessairement vrai. Il faut en tenir compte.

Remarque : Certaines équations nécessitent l'initialisation de variables intermédiaires au début de chaque période.

Pour cette raison, nous avons introduit une variable ICORK valant 1 avant le premier passage dans une équation déterminée pour chaque période, et 0 sinon.

§ 0 SOUS-ROUTINE RESOLU

Comment faire une simulation ?

Rappelons que faire une simulation de la période T1 à la période T2, c'est résoudre le modèle successivement pour les années T1 à T2, les valeurs des variables exogènes étant fixées.

Pour déterminer les valeurs des variables endogènes à la période T, nous prendrons comme point initial de la résolution au choix (guidé par la variable IRES donnée par l'utilisateur) :

- les valeurs des variables endogènes à la période T-1
- les valeurs des variables endogènes à la période T résultant d'une simulation faite antérieurement

Pour résoudre le modèle à une période T fixée, nous procédons comme suit :

- nous résolvons le prologue par l'instruction

CALL MODEL(1,L)

- nous initialisons le vecteur yb et résolvons le système réduit des équations de bouclage par la sous-routine SOL (expliquée au chapitre VII)

- nous résolvons l'épilogue par l'instruction

CALL MODEL(M+S+1,NEQUA)

La sous-routine SOL fait appel à deux autres sous-routines :

- CALFUN qui calcule la valeur des fonctions à annuler (dans notre cas E_i) au point yb
- CALJAC qui calcule le jacobien du système réduit des équations de bouclage au point yb

L'explication des deux sous-routines fera l'objet des paragraphes suivants.

§ 0 SOUS-ROUTINE CALFUN

Pour faciliter les appels aux vecteurs et aux matrices, nous introduisons les deux vecteurs suivants :

IABOU(K) contient le numéro de la variable d'assignation de la $k^{\text{ième}}$ équation de bouclage, c'est-à-dire que

$$IABOU(K) = IWATEQ(PERINI(M+K))$$

IVBOU(K) contient le numéro de la $k^{\text{ième}}$ variable de bouclage, c'est-à-dire que

$$IVBOU(K) = |IRAV(PERINI(M+K))|$$

Pour yb donné (la valeur de la $k^{\text{ième}}$ variable de bouclage à la période I se trouve dans $Y(I, IVBOU(K))$), nous résolvons d'abord le coeur en appelant $MODEL(L+1, M)$.

Ensuite, pour chaque équation de bouclage K

- nous posons $YINT = Y(I, IABOU(K))$
- nous faisons l'assignation par $CALL\ MODEL(M+K, M+K)$
- nous calculons $E(K) = Y(I, IABOU(K)) - YINT$
qui est la valeur de la fonction E_k au point yb
- nous posons $Y(I, IABOU(K)) = YINT$

§ 0 SOUS-ROUTINE CALJAC

Pour approcher le jacobien par différence finie, on peut procéder de la manière suivante :

- évaluer $E(YB)$ par un appel à CALFUN
- évaluer $E(YB + h_i v_i)$ v_i
où
 v_i est le $i^{\text{ième}}$ vecteur unité
- calculer la $i^{\text{ième}}$ colonne du jacobien
par

$$\frac{E(YB + h_i v_i) - E(YB)}{h_i}$$

Ceci nécessite $(s+1)$ appels à CALFUN,
c'est-à-dire $(s+1)$ évaluations des équations du coeur et de bouclage.

En tenant compte du creux dans la "cheminée" des variables de bouclage,
nous pouvons épargner des évaluations de fonctions de la manière suivante :

Comme dans MARIBEL la première variable de bouclage yb_1 n'intervient pas dans les 139 premières équations du coeur (équations 83 à 221) (voir figure 7), il n'est pas nécessaire de réévaluer celles-ci pour connaître les valeurs des variables correspondantes après avoir donné un accroissement h_1 à yb_1 . Par contre, un accroissement h_1 donné à la variable yb_1 , a une influence sur l'équation 222, c'est-à-dire qu'il faudra réévaluer celle-ci pour la nouvelle valeur de yb_1 .

Comme la variable yb_1 n'intervient pas non plus dans l'équation 224, on pourrait croire qu'on ne doive pas réévaluer celle-ci. Cependant, l'équation 224 en question fait intervenir la variable de l'équation 222, qui elle est modifiée lors d'un changement de la variable yb_1 . L'équation 224 est donc indirectement influencée par un accroissement h_1 donné à yb_1 ; il faudra donc la réévaluer.

Et ainsi de suite, il suffit de réévaluer tous les successeurs de y_{b_1} dans le coeur.

Les équations de bouclage à réévaluer sont celles où intervient soit y_{b_1} même, soit une variable du coeur modifiée lors d'un accroissement h_1 donné à y_{b_1} .

Après avoir évalué $E(YB + h_1 v_1)$, les variables du coeur ont donc des valeurs correspondant à $y_{b_1} + h_1$ (et non à y_{b_1}). Si nous donnons maintenant un accroissement h_2 à y_{b_2} et que nous évaluons la fonction E correspondante sans repasser par toutes les équations (procédé expliqué plus haut), nous évaluons $E(YB + h_1 v_1 + h_2 v_2)$ (et non $E(YB + h_2 v_2)$, car nous n'avons pas remis les valeurs des variables du coeur correspondant à YB). Ainsi nous utilisons automatiquement la formule (19) d'approximation par différence finie.

Comment déterminer et stocker les numéros des équations à réévaluer ?

Nous procédons en 2 étapes (étape 9 et étape 10 de la sous-routine RAVELL).

- A l'étape 9, nous cherchons pour chaque variable de bouclage les successeurs dans le coeur, puis les successeurs directs de ces variables et de la variable de bouclage en question dans les équations de bouclage.

Les équations ainsi obtenues sont celles qui sont à réévaluer lors de la détermination de la colonne du jacobien correspondant à cette variable de bouclage.

La recherche des successeurs se fait par une méthode "Branch and Bound" semblable à celle utilisée dans l'algorithme de Ravelli pour la détermination des prédécesseurs.

- A l'étape 10, nous stockons l'information ainsi obtenue dans une matrice SUCCES de façon plus accessible :
 - La colonne k de la matrice SUCCES correspond à la $k^{\text{ième}}$ variable de bouclage.

- Comme nous devons toujours évaluer des petits blocs d'équations (par exemple, pour la variable de bouclage y_{b_1} le bloc des équations 222 à 224), il ne serait pas opportun de faire un appel à MODEL pour chaque équation à évaluer.

Dans la première composante de la colonne k de SUCCES, nous stockons le nombre de blocs à évaluer dans le coeur, suivi par le début N1 et la fin N2 de chaque bloc (ce qui nous permettra de faire appel à MODEL(N1,N2)).

Comme les équations de bouclage doivent être évaluées séparément, nous stockons dans la composante suivante le nombre d'équations à réévaluer parmi celles-ci, suivi des numéros de ces dernières.

Le gain d'évaluations de fonction dans le cas de MARIBEL est impressionnant : au lieu d'évaluer $18 \times 300 = 5400$ équations, il suffit d'en évaluer 2283, c'est-à-dire qu'on obtient un gain de 58 %.

Remarques : - Tant qu'on ne change pas le vecteur IDVAR, ni l'algorithme de Ravelli, ni la détermination des successeurs ne doivent être répétés.

- Tous les résultats sur MARIBEL mentionnés au cours de ce travail concernant le problème initial de MARIBEL, c'est-à-dire le cas où les variables endogènes sont les variables d'assignation (a priori endogènes lors de la construction).

En annexe, nous fournissons une copie des sous-routines MODEL, RESOLU, CALFUN et CALJAC.

Le lecteur y trouvera également un tableau spécifiant les équations à réévaluer lors de l'approximation du jacobien.

Chapitre VI

RESOLUTION D'UNE EQUATION NON LINEAIRE A UNE INCONNUE

§ 0 POSITION DU PROBLÈME

Soit $f : R \rightarrow R$
 $x \rightarrow f(x)$

une fonction réelle de classe C^n .

Il s'agit de résoudre l'équation :

$$(62) \quad f(x) = 0$$

Un processus itératif de résolution de (62) est en général défini par la donnée d'une ou de plusieurs approximations initiales x_0, x_1, \dots, x_p et d'un algorithme

$$x_k = A(x_{k-1}, x_{k-2}, \dots, x_{k-p-1})$$

où A est une application de R^{p+1} dans R .

On dit que l'algorithme A converge si, lorsque k augmente indéfiniment, la suite (x_k) des itérés successifs converge dans R vers une des solutions x^* du problème (62).

On note e_k l'erreur de l'algorithme A à l'itération k .

Nous avons :

$$e_k = |x_k - x^*|$$

Nous disons que la convergence de l'algorithme est d'ordre p ssi

$$\exists c \in R_0^+ \text{ tel que } \lim_{k \rightarrow \infty} \frac{e_{k+1}}{(e_k)^p} = c$$

Remarque : L'ordre de la convergence est une notion locale.

Nous appelons itération de l'algorithme A le passage de l'approximation x_{k-1} du point x^∞ à l'approximation x_k .

L'algorithme A est d'autant plus efficace que :

- c est petit
- p est grand
- le coût d'une itération est peu élevé

En général, on donne un intervalle $[a, b]$ délimitant la recherche de la racine.

Il est important que l'algorithme s'arrête en donnant une information adéquate s'il n'existe pas de solution dans l'intervalle $[a, b]$.

Dans le cas où il y aurait plusieurs solutions dans l'intervalle $[a, b]$, il est souhaitable que l'algorithme converge vers la solution la plus proche de x_0 .

Nous allons décrire ci-dessous plusieurs méthodes itératives de résolution du problème (62).

§ 0 QUELQUES MÉTHODES DE RÉOLUTION D'UNE ÉQUATION NON LINÉAIRE

Remarque : Les hypothèses et conclusions des différentes méthodes ne sont données qu'approximativement.

En effet, le but de ce chapitre n'est pas de faire une théorie approfondie, mais de passer en revue les différentes méthodes existantes.

Pour une théorie mathématiquement fondée, voir <9> et <10>.

a) La bisection

Hypothèses : • Il existe au plus une racine dans l'intervalle $[a, b]$

• $f(a) \cdot f(b) < 0$

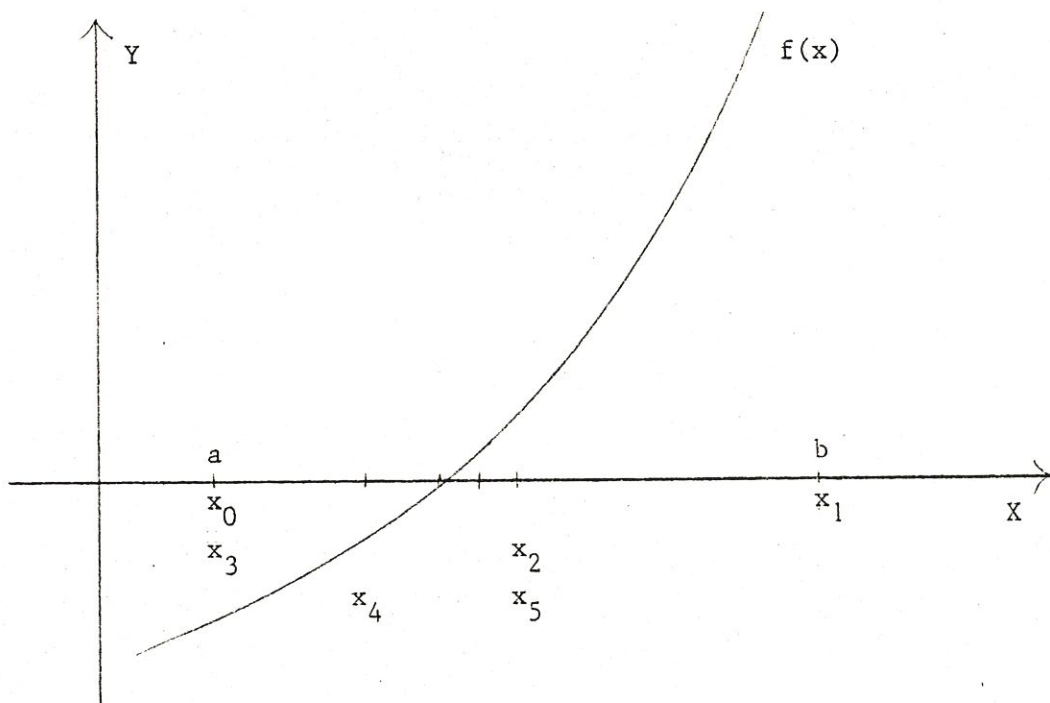
• f continue

Posons $x_0 = a$

$x_1 = b$

Calculons $x_{2k} = \frac{x_{2k-1} + x_{2k-2}}{2}$

$$x_{2k+1} = \begin{cases} x_{2k-1} & \text{si } f(x_{2k-1}) \cdot f(x_{2k}) < 0 \\ x_{2k-2} & \text{si } f(x_{2k-2}) \cdot f(x_{2k}) < 0 \end{cases}$$



Avantages : • programmation très facile

• coût d'une itération peu élevé

Inconvénients : • hypothèses restrictives

• convergence linéaire (i.e. d'ordre 1) ; $c = \frac{1}{2}$

b) Méthodes d'interpolation polynomiale d'ordre n

Hypothèses : • f continue

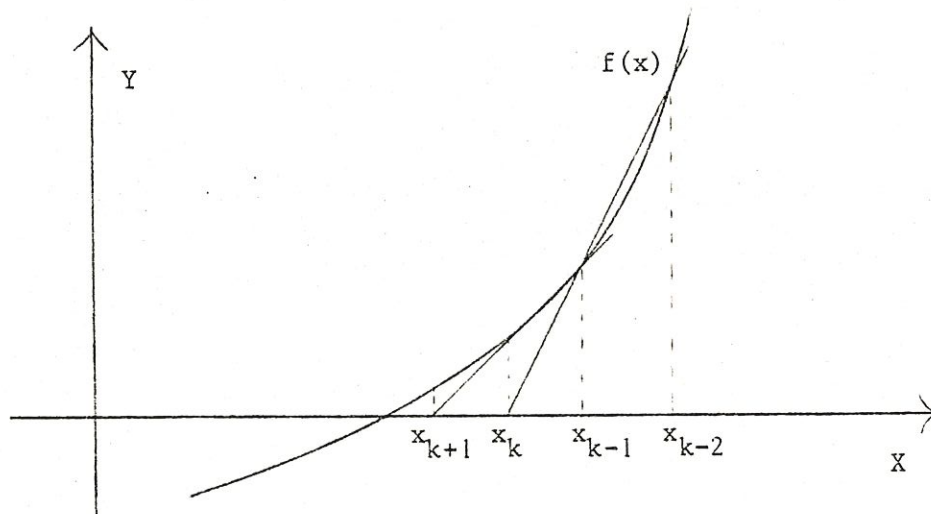
• connaissance de $(n+1)$ approximations initiales : x_0, \dots, x_n

A l'itération k , on construit le polynôme interpolant d'ordre n , $P_n(x)$, tel que

$$P_n(x_{k-j}) = f(x_{k-j}) \quad \text{pour } j=1, \dots, n+1$$

x_k est une racine du polynôme P_n .

Cas $n = 1$: Cette méthode se ramène alors à la méthode bien connue de la
=====
sécante.



Avantages : • convergence superlinéaire pour $f \in C^2$

l'ordre de convergence p_n tend vers 2 quand n augmente :

$$p_1 = 1.62$$

$$p_2 = 1.84$$

$$p_3 = 1.93$$

$$p_4 = 1.94$$

• convergence plus ou moins globale

Inconvénients : • pour $n > 2$, le calcul de P_n et la recherche de ses racines sont coûteux

• pour $n > 1$, la racine n'est pas unique et peut être complexe

c) Méthode de Newton

Hypothèses : • f de classe C^1

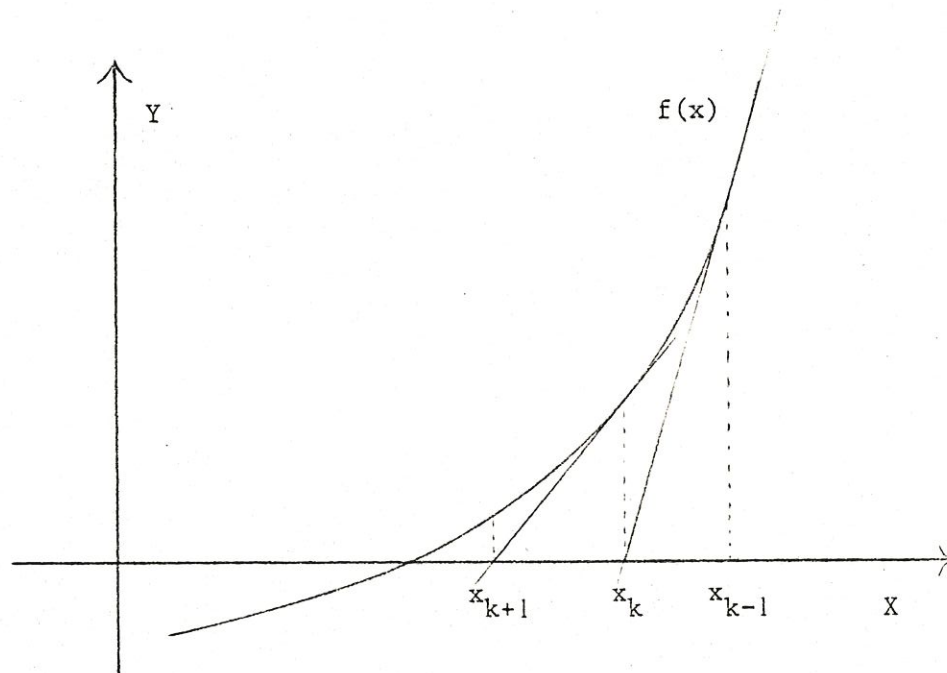
- connaissance de $f'(x)$ $\forall x \in [a, b]$
- connaissance d'une approximation initiale x_0
- $f'(x) \neq 0$ $\forall x \in [a, b]$

A l'itération k , on approche la fonction f par son développement de Taylor d'ordre 1 au point x_{k-1} (c'est-à-dire par sa tangente).

Le point d'intersection de la tangente avec l'axe des X donne la prochaine approximation x_k de la solution.

Nous avons :

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})}$$



Avantages : • convergence quadratique si f de classe C^2

- programmation facile
- coût d'une itération peu élevé

Inconvénients : • convergence fort locale

- connaissance nécessaire de f'

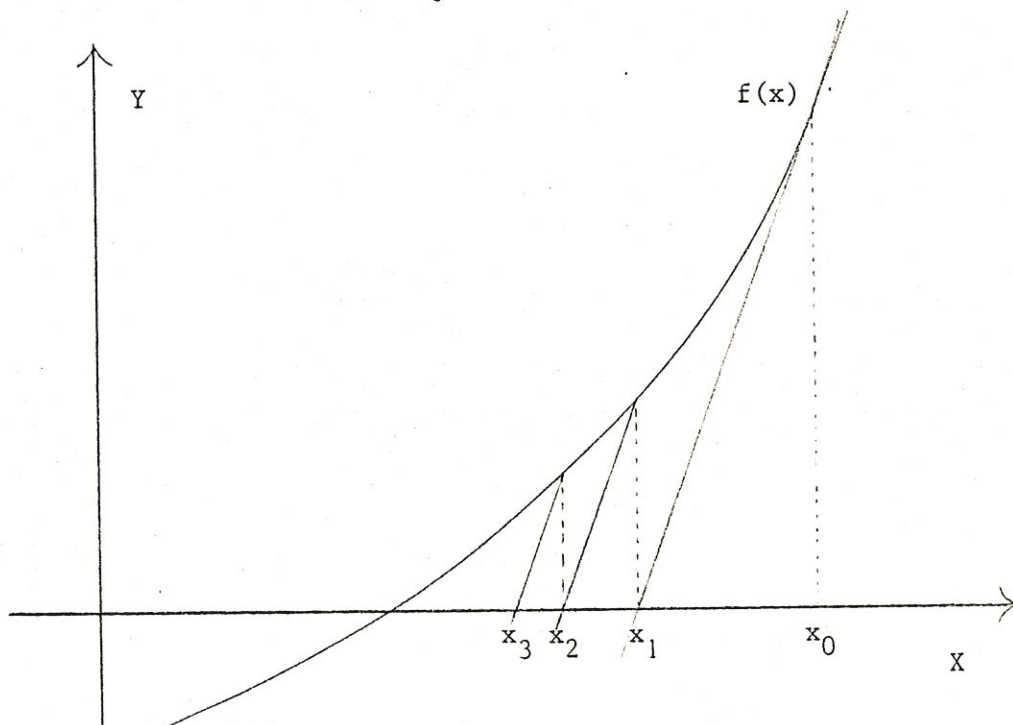
d) Méthode de Newton modifiée

Hypothèses : • f de classe C^1

- connaissance d'une approximation initiale x_0
- connaissance de $f'(x_0)$
- $f'(x_0) \neq 0$

L'itération est définie par :

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_0)}$$



Avantages : • programmation facile

• coût d'une itération peu élevé

• par rapport à la méthode de Newton : connaissance de $f'(x_0)$ uniquement

Inconvénients : • convergence fort locale

• convergence linéaire

Remarque : Une autre variante consiste à évaluer $f'(x)$ seulement toutes les m itérations.

e) Méthode de Quasi-Newton

Hypothèses : • f de classe C^1

• connaissance d'une approximation initiale x_0

$$\text{Soit } f'_h = \frac{f(x+h) - f(x)}{h}$$

L'itération est donnée par :

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'_{h_{k-1}}(x_{k-1})}$$

La méthode Quasi-Newton n'est rien d'autre que celle de Newton dans laquelle on approche la dérivée par différence finie.

Avantages : • convergence superlinéaire si $f \in C^2$

• si $\lim_{k \rightarrow \infty} h_k = 0$, convergence quadratique pour $f \in C^2$

• par rapport à la méthode de Newton, la connaissance de f' n'est pas nécessaire

Inconvénients : • convergence fort locale

• deux évaluations de f par itération

f) Approximation du second degré

Hypothèses : • f de classe C^2

• connaissance de $f'(x)$

$$\forall x \in [a, b]$$

et de $f''(x)$

• connaissance d'une approximation initiale x_0

$$\bullet f'^2(x) - 2 \cdot f(x) \cdot f''(x) \geq 0$$

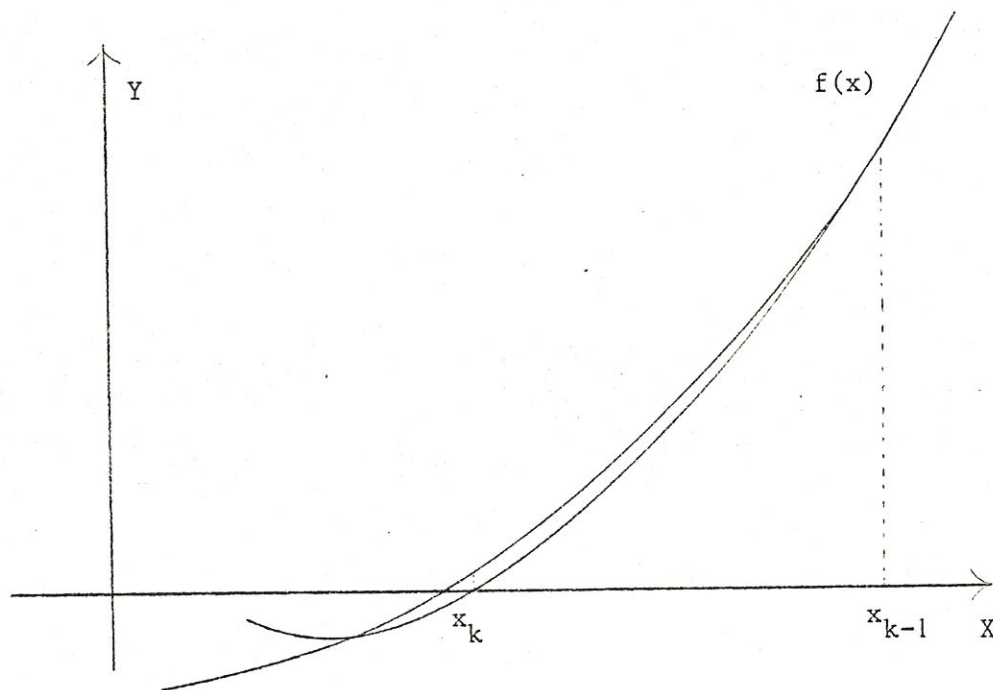
$$\forall x \in [a, b]$$

A l'itération k , on approche la fonction par son développement de Taylor d'ordre 2.

Le point d'intersection de cette approximation avec l'axe des X , le plus proche de x_{k-1} , donne la prochaine approximation x_k de la solution.

Nous avons :

$$x_k = x_{k-1} - \frac{2 \cdot f(x_{k-1})}{f'(x_{k-1}) + \text{signe}(f'(x_{k-1})) \cdot \sqrt{f'(x_{k-1})^2 - 2 \cdot f(x_{k-1}) \cdot f''(x_{k-1})}}$$



Avantages :

- convergence cubique pour f de classe C^3
- convergence plus globale que la méthode de Newton

Inconvénients :

- connaissance nécessaire de f' et f''
- hypothèse restrictive : $f'^2 - 2 \cdot f \cdot f'' \geq 0$
(parfois x_0 doit être très près de x^* pour que cette hypothèse soit vérifiée)

g) Variante de l'approximation du second degré

Hypothèses : • f de classe C^2

• connaissance d'une approximation initiale x_0

• $f_h'^2 - 2 \cdot f \cdot f_h'' \geq 0$

Soient

$$(63) \quad f_h'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f_h''(x)$$

$$(64) \quad f_h''(x) = \frac{f(x+2h) - 2f(x+h) + f(x)}{h}$$

L'itération s'écrit :

$$x_k = x_{k-1} - \frac{2 \cdot f(x_{k-1})}{f_h'(x_{k-1}) + \text{signe}(f_h'(x_{k-1})) \sqrt{f_h'(x_{k-1})^2 - 2 \cdot f(x_{k-1}) \cdot f_h''(x_{k-1})}}$$

Nous avons remplacé f' et f'' par leurs approximations dans la formule précédente.

Avantages : • si $\lim_{k \rightarrow \infty} h_k = 0$, convergence cubique pour f de classe C^3

• par rapport à la méthode précédente, la connaissance de f' et de f'' n'est pas nécessaire

Inconvénients : • hypothèse restrictive : $f_h'^2 - 2 \cdot f \cdot f_h'' \geq 0$

• trois évaluations de f par itération

§ 0 ALGORITHME UTILISÉ

L'algorithme de résolution d'une équation non linéaire à une inconnue utilisé dans MODEL se base sur la méthode proposée en g).

- La première difficulté de la méthode proposée en g) est la nécessité de l'hypothèse

$$(65) \quad f'_h(x_{k-1})^2 - 2 f(x_{k-1}) f''_h(x_{k-1}) \geq 0$$

qui n'est pas toujours réalisée.

Dans le cas où (65) n'est pas vérifiée, nous procédons comme suit:

Nous approchons la fonction f par un polynôme $P(x)$ de degré 2 tel que

$$P(x_{k-1}) = f(x_{k-1})$$

$$P'(x_{k-1}) = f'_h(x_{k-1})$$

$P(x)$ admette un zéro double

On vérifie facilement que le polynôme

$$P(x) = f(x_{k-1}) + f'_h(x_{k-1}) (x - x_{k-1}) + \frac{f'_h(x_{k-1})^2}{2 f(x_{k-1})} \frac{(x - x_{k-1})^2}{2}$$

satisfait les trois conditions citées plus haut.

Remarquons que la dérivée seconde de $P(x)$ vaut :

$$\frac{f'_h(x_{k-1})^2}{2 f(x_{k-1})}$$

Le pas à effectuer $(x_k - x_{k-1})$ sera alors la racine double du polynôme $P(x)$,

c'est-à-dire que

$$x_k - x_{k-1} = - \frac{2 f(x_{k-1})}{f'_h(x_{k-1})}$$

Notons que le pas ainsi défini est le double pas de Newton.

- En pratique, l'approximation de la dérivée seconde f''_h par la formule (64) est parfois mauvaise. En effet, lors du calcul des différences, on perd facilement des chiffres significatifs (notamment quand h est choisi trop petit).

Le problème est maintenant de trouver un critère permettant de décider que l'approximation de la dérivée seconde est "convenable".

Nous utiliserons le critère suivant :

Si la correction

$$- \frac{h}{2} f''_h(x)$$

apportée à la première approximation

$$\frac{f(x+h) - f(x)}{h}$$

de la dérivée première fait changer celle-ci de signe, nous estimons que la valeur de $f''_h(x)$ est mauvaise.

En effet, nous pouvons admettre que

$$\frac{f(x+h) - f(x)}{h}$$

est une meilleure approximation de $f'(x)$ que $f''_h(x)$ ne l'est de la dérivée seconde, et dès lors que le critère est valable.

Dans le cas où la dérivée seconde est mauvaise (au sens du critère cité plus haut), nous posons :

$$f''_h(x) = 0$$

Ceci revient à utiliser le pas de Newton.

- A l'itération k , nous appelons pas prévu, noté PREVU, le pas $(x_k - x_{k-1})$ où x_k annule le polynôme $P(x)$ approchant la fonction f au point x_{k-1} .

Le pas prévu est donné par la formule :

$$(66) \quad x_k - x_{k-1} = \frac{-2 \cdot f(x_{k-1})}{f'_h(x_{k-1}) + \text{signe}(f'_h(x_{k-1})) \cdot \sqrt{f'_h(x_{k-1})^2 - 2 \cdot f(x_{k-1}) \cdot f''_h(x_{k-1})}}$$

si $f'_h(x_{k-1})^2 - 2 \cdot f(x_{k-1}) \cdot f''_h(x_{k-1}) \geq 0$

Remarquons que cette formule donne bien le pas de Newton quand $f''_h(x_{k-1}) = 0$ (c'est-à-dire quand l'approximation de la dérivée seconde est "mauvaise").

$$(67) \quad x_k - x_{k-1} = - \frac{2 \cdot f(x_{k-1})}{f'_h(x_{k-1})} \quad \text{sinon}$$

- Le caractère local de la convergence de la méthode de Newton est essentiellement dû au fait que le pas devient énorme dès que la dérivée est petite par rapport à la fonction.

Par exemple, pour la fonction

$$f(x) = \ln x$$

la méthode de Newton donne

$$x_k = x_{k-1} - x_{k-1} \cdot \ln x_{k-1}$$

Or si $x_{k-1} > e$, alors $x_k < 0$.

Il s'en suit que le pas prévu est trop grand.

Un raisonnement analogue peut se faire pour la méthode par approximation du second degré.

Pour ces raisons, nous allons essayer de contrôler la longueur du pas à effectuer.

Nous introduisons une variable supplémentaire DELTA, ajustée à chaque itération, qui est une borne supérieure du pas à effectuer réellement.

Ainsi

- si la longueur du pas prévu est plus petite ou égale à DELTA, nous effectuons le pas prévu
- sinon, nous faisons un pas de longueur DELTA dans la direction du pas prévu.

A la première itération, DELTA est initialisé à la longueur du pas prévu.

Examinons maintenant la révision de DELTA.

Dans ce qui suit, nous travaillons avec la valeur absolue de la fonction f , soit F .

A l'itération k , nous appelons PHI la valeur prévue de la fonction F au point $x_k = x_{k-1} + \text{PAS}$
où PAS est le pas réel effectué.

PHI s'obtient par la formule suivante :

$$\text{PHI} = \left| P(x_k) \right|$$

La diminution prévue de la valeur de la fonction F vaut

$$F(x_{k-1}) - \text{PHI}$$

tandis que la diminution réelle est donnée par

$$F(x_{k-1}) - F(x_k)$$

Introduisons la variable RAPPOR

telle que

$$\text{RAPPOR} = 1 - \frac{F(x_{k-1}) - F(x_k)}{F(x_{k-1}) - \text{PHI}}$$

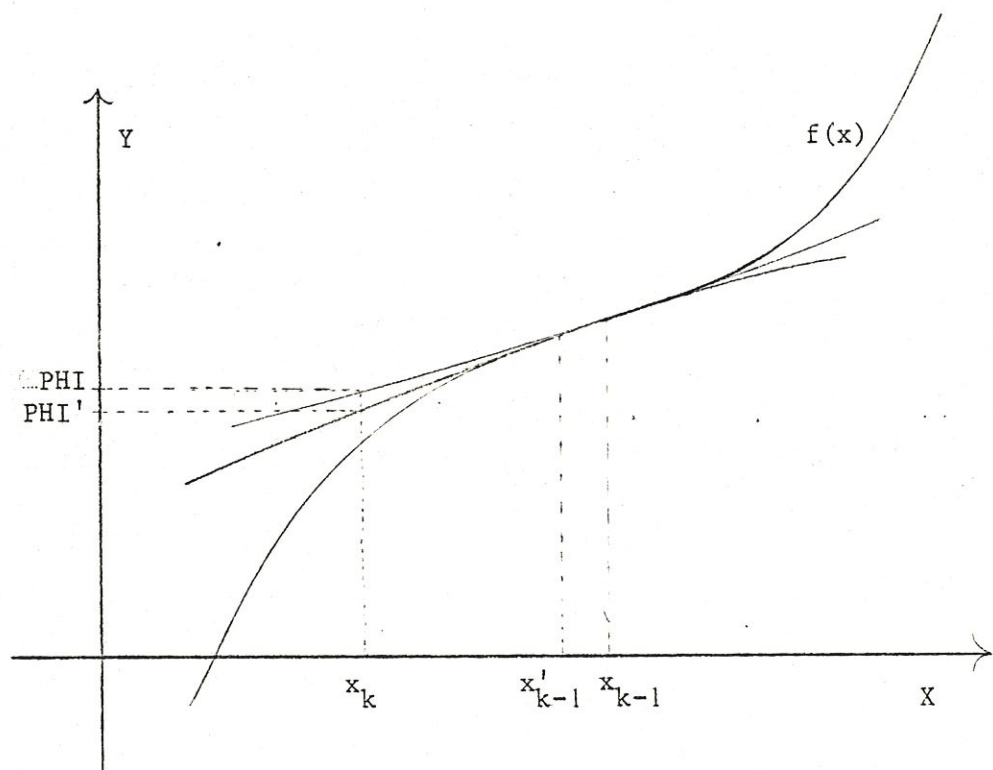
Plusieurs cas se présentent :

1) RAPPOR < 0

signifie que la diminution réelle est plus grande que la diminution prévue.

Dans ce cas, nous ne pouvons dire si l'approximation de la fonction f par le polynôme P dans l'intervalle $[x_{k-1}-\text{DELTA}, x_{k-1}+\text{DELTA}]$ est bonne ou non.

En effet, les deux situations peuvent se présenter :



La valeur de PHI , et donc de RAPPOR , est presque la même dans les deux cas.

Nous ne pouvons donc estimer la validité de l'approximation de f par P sur l'intervalle $[x_{k-1}-\text{DELTA}, x_{k-1}+\text{DELTA}]$, c'est-à-dire que nous ne savons pas s'il faut augmenter ou diminuer DELTA .

Dans ce cas, nous ne changeons pas la valeur de DELTA .

2) RAPPOR = 0

signifie que la diminution réelle est égale à la diminution prévue.

Il est probable que l'approximation est bonne sur l'intervalle en question et qu'elle le sera encore sur un intervalle plus étendu.

Nous décidons de doubler la valeur de DELTA.

3) $0 < \text{RAPPOR} < 1/2$

signifie que la diminution réelle se situe entre 50 % et 100 % de la diminution prévue.

Nous estimons qu'on peut augmenter DELTA d'un facteur LAMBDA (qui diminue lorsque RAPPOR augmente).

4) $\text{RAPPOR} > 1/2$

signifie qu'on ne diminue même pas de la moitié de la diminution prévue.

Dans ce cas, nous considérons que l'intervalle de validité de l'approximation est trop grand. Il s'en suit que nous diminuons DELTA.

Même si RAPPOR devient très grand, nous imposons que le facteur de multiplication de DELTA reste supérieur à $1/4$.

En résumé, si LAMBDA est le facteur de multiplication, le graphique exprimant LAMBDA en fonction de RAPPOR, aura l'allure représentée à la figure suivante.

Nous avons imposé :

$$\text{LAMBDA}(0) = 2$$

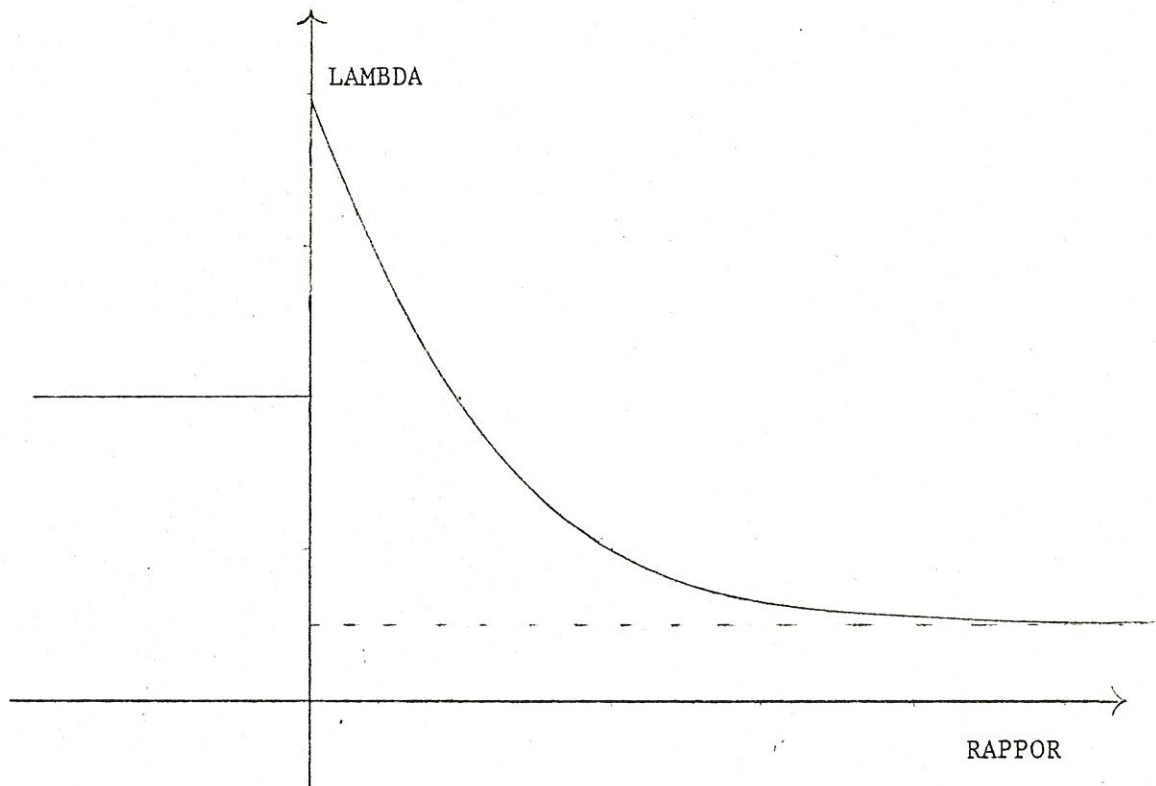
$$\text{LAMBDA}(1/2) = 1$$

$$\text{LAMBDA}(1) \approx 1/2$$

$$\text{LAMBDA}(\infty) = 1/4$$

$$\text{LAMBDA}(\text{RAPPOR}) = 1$$

$$\forall \text{ RAPPOR} < 0$$



Le lecteur constatera que la fonction

$$\text{LAMBDA}(\text{RAPPOR}) = \begin{cases} -1 & \text{si } \text{RAPPOR} < 0 \\ 1.75 e^{b \cdot \text{RAPPOR}} + 0.25 & \text{si } \text{RAPPOR} > 0 \end{cases}$$

avec $b = 2 \ln \frac{3}{7}$

répond à la question.

Cependant, nous ne choisissons pas cette fonction, car l'appel d'une exponentielle à chaque itération est relativement cher.

Nous approchons la fonction comme suit :

$$\text{LAMBDA}(\text{RAPPOR}) = \begin{cases} 2 (1 - \text{RAPPOR}) & \text{si } 0 < \text{RAPPOR} < 0.5 \\ \frac{11}{8} - \frac{3}{4} \text{RAPPOR} & \text{si } 0.5 < \text{RAPPOR} < 1.5 \\ 0.25 & \text{si } 1.5 < \text{RAPPOR} \end{cases}$$

L'expérience montre qu'il ne faut pas augmenter DELTA trop rapidement.

Nous imposons qu'il faut LAMBDA plus grand que 1 lors de deux itérations successives pour augmenter DELTA.

- Comme les dérivées première et seconde sont approchées par les formules (63) et (64), il faut déterminer la valeur du pas h utilisé.

Nous posons

$$h = \begin{cases} \text{DSTEP} & \text{si } x \leq 1 \\ \text{DSTEP} \cdot x & \text{si } x > 1 \end{cases}$$

où DSTEP est donné par l'utilisateur.

Nous utilisons la valeur de 10^{-6} pour DSTEP.

- Vu le fait que les directions calculées sont des directions de descente, nous estimons que la valeur de la fonction doit diminuer quand on fait un pas "assez petit" dans une de ces directions.
Si la fonction ne décroît pas avec un pas plus petit que DSS, l'algorithme s'arrête en signalant une erreur.
Le paramètre DSS donné par l'utilisateur, vaut 10^{-12} dans nos programmes.
- L'algorithme s'arrête également si le nombre maximum d'appels de fonction permis, MAXFUN, est atteint.
- L'utilisateur doit fournir le paramètre DMAX qui est une approximation généreuse de la distance du point initial x_0 au point solution x^* .
Si, à une itération donnée k , le pas prévu est plus grand que DMAX, l'algorithme s'arrête en indiquant un point stationnaire.
En effet, dans ce cas il est fort probable que la fonction admette un minimum local dans le voisinage de x_{k-1} sans s'annuler.
- Lors de la révision de DELTA, nous imposons une borne inférieure et supérieure à DELTA valant respectivement DSS et DMAX.

En résumé, l'algorithme peut être décrit comme suit :

Etape 1

Calcul des approximations des dérivées première et seconde par les formules (63) et (64).

Calcul du pas prévu par les formules (66) ou (67).

Etape 2

Si $PREVU < DELTA$, poser $PAS = PREVU$

$PHI = 0$

Sinon : $PAS = \text{signe}(PREVU) \cdot DELTA$

$PHI = P(x + PAS)$

Etape 3

Calcul de $f(x + PAS)$

Révision de $DELTA$

Etape 4

Test d'échange

Si $f(x + PAS) < f(x)$, poser $x = x + PAS$

aller à l'étape 1

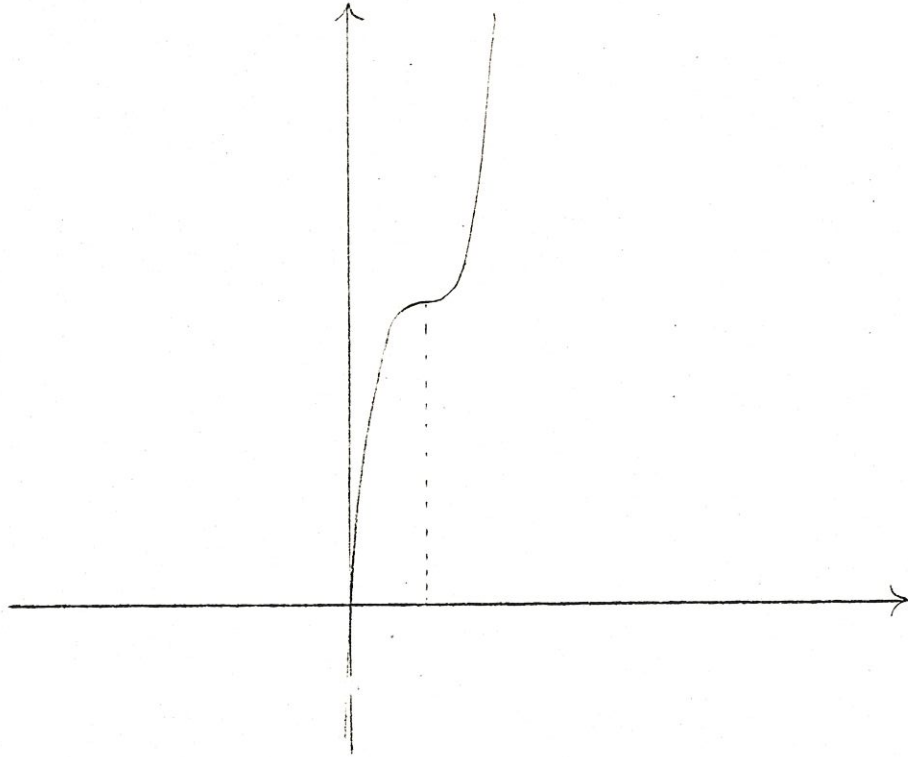
Sinon, aller à l'étape 2.

Expliquons le fonctionnement de DELTA à l'aide d'un exemple :

Soit la fonction

$$f(x) = (x - 2)^3 + 8$$

L'allure de la fonction est la suivante :



Il est facile de voir que la fonction admet une racine unique en 0 et un point de selle en 2.

Ce point de selle peut poser problème à des algorithmes qui ne tiennent compte que des propriétés locales de la fonction.

Voyons comment se comporte notre algorithme.

- Si le point initial se trouve près de 2, le pas prévu sera très grand. Dès lors, suivant la valeur de DMAX, l'algorithme va effectuer un pas très grand ou s'arrêter en signalant un point stationnaire.

Vu le fait que nous n'avons qu'une information locale (c'est la première itération), ce comportement ne peut être empêché.

Voyons ce qui se passe pour $x_0 = 2.788$

Les résultats des itérations successives sont celles-ci :

```

      0.
      2.788
    8.489303872
      1.86283184
    4.729688005

      1.
    -6.326407097      N
    -569.2619369

      2.
    .5093982257      N †
    4.688041383
    6.665680717
    -8.943045331

      3.
    -0.011731317      Δ²
    -.1416031607
    12.14118863
    -12.0695

      4.
    .0000001295      Δ²
    .0000015541
    11.99999847
    -11.9997

      5.
    -5.9681803-12      Δ²

LA SOLUTION FINALE
EST:

    -5.9681803-12      X
      -4. -11 F(X)

```

Les 5 chiffres dans un bloc sont dans l'ordre :

- le numéro de l'itération
- la valeur de x
- la valeur de $f(x)$
- la valeur de $f'_h(x)$
- la valeur de $f''_h(x)$

Le symbole N signifie qu'on a utilisé un double pas de Newton, tandis que le symbole Δ^2 indique que le pas utilisé est calculé par la formule (66).

Le symbole † indique que le pas prévu est rejeté (car trop grand) et qu'on utilise un pas de longueur DELTA.

Pour cet exemple, nous avons mis DMAX à 10.

A la première itération, le pas prévu est très grand, mais vu le fait que DMAX vaut 10 et que DELTA est initialisé à la longueur du pas prévu, l'algorithme effectue le pas prévu sans succès (la fonction ne décroît pas).

Ceci entraîne que DELTA est diminué.

Le pas prévu est toujours le même, mais maintenant sa norme est plus grande que DELTA. Le pas prévu est alors rejeté et l'algorithme effectue un pas de longueur DELTA.

Si nous choisissons 5.6 comme point de départ, le déroulement de l'algorithme est le suivant :

```

0.
  5.6
  54.656
38.87999936
21.6033801

```

```

1.
  2.78847732
  8.490193577
  1.865089287
  4.732557324

```

N

```

2.
-0.0230453605
-0.2797430965
12.27813758
-12.1377

```

N ↑

```

3.
.00000010074
.00000120892
11.99998792
-11.9995

```

Δ²

```

4.
-1.93796-13 Δ2

```

LA SOLUTION FINALE
EST :

```

-1.93796-13 X
0. F(X)

```

Le point x_1 trouvé à partir de x_0 est le point initial du problème précédent.

Dans ce cas, nous possédons déjà l'information de l'itération précédente (contenue en quelque sorte dans DELTA) et le pas prévu (toujours le même qu'au problème précédent) est tout de suite rejeté et est remplacé par un pas de longueur DELTA.

Remarquons cependant que le contrôle du pas à effectuer peut aussi ralentir la convergence pour certaines fonctions.

Appliquons l'algorithme à la fonction suivante :

$$f(x) = \frac{1}{x} - \frac{1}{10}$$

en partant près du point singulier 0, soit $x_0 = \frac{1}{10}$

Voici les différentes itérations de l'algorithme :

0.		7.	
0.1		3.293670911	N ↑
9.9		.2036126034	
-99.999998		-0.092180611	
1999.4		.0559573193	
1.		8.	
0.298000004	N	4.835251235	N ↑
3.255704653		.1068144862	
-11.26075341		-.0427722305	
75.55064886		.0176858901	
2.		9.	
.4960000079	N ↑	7.158070144	N ↑
1.916129		.0397024589	
-4.064776034		-.0195167764	
16.38470616		0.005451231	
3.		10.	
.7439624145	N ↑	10.76131101	N ↑
1.244153926		-.0070745192	
-1.806749728		-.0086351448	
4.855459348		.0016044099	
4.		11.	
1.078208996	N ↑	9.99639588	Δ^2
.8274639738		.0000360542	
-.8601893978		-0.010007212	
1.595049246		.0020016426	
5.		12.	
1.556323757	N ↑	10.	Δ^2
.5425398284			
-0.412857418			
.5303649131			
6.			
2.257359161	N ↑		
0.342995522			
-.1962450281			
.1738181502			

LA SOLUTION FINALE
EST:

10. X
-1.7-12 F(X)

Vu le fait que le pas prévu à la première itération est relativement petit, la valeur initiale de DELTA est également petite et comme la décroissance de f est très faible, nous n'augmentons DELTA que difficilement.

Chapitre VII

RESOLUTION D'UN SYSTEME D'EQUATIONS NON LINEAIRES

Dans ce chapitre, nous donnons un aperçu bref de quelques méthodes existantes de résolution d'un système d'équations non linéaires ainsi qu'une description plus détaillée de l'algorithme choisi pour résoudre le système réduit des s équations de bouclage.

Soit le système d'équations défini par

$$(68) \quad f(x) = 0$$

avec $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$

ou

$$(69) \quad f_i(x) = 0$$

avec $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$

Les considérations générales faites au chapitre précédent, concernant la définition d'un algorithme, de sa convergence, etc... s'étendent facilement au cas d'une fonction de \mathbb{R}^n dans \mathbb{R}^n .

§ 0 MÉTHODE DE NEWTON ET SES VARIANTES

La méthode de Newton et ses variantes se généralisent sans trop de peine au cas d'un système de dimension n , en remplaçant la dérivée $f'(x)$ par le jacobien $J(x)$ du système (ou par son approximation).

Remarquons qu'une itération de la méthode de Newton discrétisée risque de devenir assez coûteuse vu le fait que l'approximation du jacobien par différence finie nécessite $(n+1)$ évaluations de la fonction f .

De plus, le calcul du pas $(x_k - x_{k-1})$ par la formule

$$J(x_{k-1}) (x_k - x_{k-1}) = f(x_{k-1})$$

exige la résolution d'un système linéaire à chaque itération.

Pour remédier à ces difficultés, Broyden propose des formules de mise à jour pour l'approximation du jacobien et de son inverse à chaque itération.

Les formules de correction de rang 1,

$$(70a) \quad J_k = J_{k-1} + \frac{(\gamma_{k-1} - J_{k-1} \delta_{k-1}^*) \delta_{k-1}^*}{\|\delta_{k-1}\|^2}$$

$$(70b) \quad H_k = H_{k-1} + \frac{(\delta_{k-1} - H_{k-1} \gamma_{k-1}) \delta_{k-1}^* H_{k-1}}{\delta_{k-1}^* H_{k-1} \gamma_{k-1}}$$

où J_k est l'approximation du jacobien

H_k est l'inverse de J_k

$$\delta_{k-1} = x_k - x_{k-1}$$

$$\gamma_{k-1} = f(x_k) - f(x_{k-1})$$

proposées par Broyden, ont la signification intuitive qui suit :

La dérivée directionnelle de la fonction dans la direction $(x_k - x_{k-1})$ est approchée par le rapport

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

En d'autres mots, nous avons ajusté l'approximation J_{k-1} de telle manière que

$$J_k (x_k - x_{k-1}) = f(x_k) - f(x_{k-1})$$

Pour l'approximation initiale de J, on peut choisir soit la matrice d'identité, soit l'approximation par différence finie de la matrice jacobienne au point initial.

Broyden a montré que la convergence de la méthode reste superlinéaire.

Cependant cette méthode a deux inconvénients majeurs :

- Le dénominateur de la formule (70b) peut devenir très petit ce qui entraîne des instabilités numériques.

On peut y remédier en introduisant un scalaire α de la façon suivante :

$$(71a) \quad J_k = J_{k-1} + \frac{\alpha (\gamma_{k-1} - J_{k-1} \delta_{k-1}^*) \delta_{k-1}^*}{\|\delta_{k-1}\|^2}$$

$$(71b) \quad H_k = H_{k-1} + \frac{\alpha (\delta_{k-1} - H_{k-1} \gamma_{k-1}) \delta_{k-1}^* H_{k-1}}{\alpha (\delta_{k-1}^* H_{k-1} \gamma_{k-1}) + (1-\alpha) \|\delta_{k-1}\|^2}$$

$$\text{où} \quad \alpha = \begin{cases} 0.8 & \text{si } |\delta_{k-1}^* H_{k-1} \gamma_{k-1}| < 0.1 \|\delta_k\|^2 \\ 1 & \text{sinon} \end{cases}$$

Remarquons que pour $\alpha = 1$, les formules se ramènent aux formules (70).

- Comme mentionné plus haut, les ajustements du jacobien se font toujours dans la direction du pas effectué. Il s'en suit que l'approximation du jacobien perd sa validité dans les directions non engendrées par les pas effectués.

Powell propose le procédé suivant pour éviter la perte d'informations dans les directions non utilisées :

On exige que les pas (utilisés pour l'ajustement du jacobien) soient "suffisamment" linéairement indépendants (dans le sens que les m derniers pas doivent engendrer l'espace tout entier), faute de quoi on utilise un "extra-step" dans la direction "manquante" pour la mise à jour du jacobien qui ne se base dès lors pas sur le pas effectué.

Pour plus de détails, nous renvoyons à Powell (<8>).

§ 0 MÉTHODE DE BROWN

Cette méthode combine la méthode de Newton unidimensionnelle avec l'élimination de Gauss pour la résolution des systèmes linéaires. Partant d'une approximation x_0 , on linéarise la première équation

$$(72) \quad f_1(x_0) + \sum_{i=1}^n \frac{\partial f_1(x_0)}{\partial x^i} (x^i - x_0^i) = 0$$

On cherche i tel que

$$\left| \frac{\partial f_1(x_0)}{\partial x^i} \right|$$

soit le plus grand possible.

Supposons, pour simplifier l'exposé, que ce soit x^1 .

On résoud en x^1 l'équation (72) et l'on obtient :

$$(73) \quad x^1 = h_1(x^2, x^3, \dots, x^n)$$

$$= x_0^1 - \left(\frac{\partial f_1(x_0)}{\partial x^1} \right)^{-1} \left(f_1(x_0) + \sum_{i=2}^n \frac{\partial f_1(x_0)}{\partial x^i} (x^i - x_0^i) \right)$$

On définit ainsi $n-1$ nouvelles fonctions :

$$(74) \quad f_{i,2}(x^2, x^3, \dots, x^n) = f_i \left(h_1(x^2, x^3, \dots, x^n), x^2, x^3, \dots, x^n \right) \\ i=2, \dots, n$$

Le problème est maintenant de trouver x^2, x^3, \dots, x^n vérifiant $f_{i,2} = 0, i \geq 2$.
On élimine alors la variable de numéro i telle que

$$\left| \frac{\partial f_{2,2}}{\partial x^i} \right|$$

soit le plus grand, supposons que ce soit x_2 pour simplifier.

On aboutit à un système $f_{i,3} = 0, i \geq 3$ en x^3, \dots, x^n .

Et ainsi de suite jusqu'à obtenir un système unidimensionnel :

$$(75) \quad f_{n,n}(x^n) = 0$$

dont la solution x_1^n est une nouvelle approximation de $(x^n)^*$.

On "remonte" ensuite les calculs en calculant les x_1^i récursivement :

$$(76) \quad x_1^i = h_i(x_1^{i+1}, \dots, x_1^n) \quad i=n-1, n-2, \dots, 1$$

et on répète le processus à partir de x_1 .

Brown a démontré la convergence localement quadratique du processus pour les fonctions de classe C^2 .

Nous renvoyons à Brown (<21>) pour une description plus détaillée de la méthode et pour des tests de comparaison avec d'autres méthodes.

Pour la linéarisation (bien sûr par différence finie) de l'équation i , il faut $(n+2-i)$ appels à la fonction f_i .

En tout, il faudra donc

$$\frac{n^2 + 3n}{2} \text{ appels de fonction } f_i.$$

Comparé aux $(n^2 + n)$ appels de fonction nécessaire pour le calcul du jacobien par différence finie, la méthode de Brown paraît plus avantageuse que celle de Newton (sans utilisation des formules de Broyden).

Cependant le coût des calculs intermédiaires à une itération est plus élevé.

La méthode de Brown semble donc être avantageuse pour des problèmes où le coût d'un appel de fonction est très élevé.

L'algorithme pourrait donc être intéressant pour la résolution de modèles macro-économiques.

Mais un inconvénient se présente lors de l'application de cette méthode au système réduit des s équations de bouclage. Rappelons que pour évaluer une fonction E_i de bouclage en un point y_b , il faut passer à travers le coeur (m évaluations de fonction) et ensuite évaluer l'équation de bouclage en question. Il s'en suit que le coût de calcul d'un élément (i,j) de la matrice jacobienne est presque aussi élevé que le coût de calcul de toute la $j^{\text{ième}}$ colonne du jacobien (pour le dernier, il ne faut que $s-1$ évaluations supplémentaires et généralement s est petit par rapport à m).

Vu les transformations effectuées sur les fonctions (f_i devient $f_{i,j}$) dans la méthode de Brown, on ne peut profiter de cette structure.

§ 9 MÉTHODE DE GAUSS-SEIDEL

Il existe de nombreuses variantes de cette méthode, mais nous nous contentons d'en donner l'idée de base (en renvoyant le lecteur intéressé à Orthega-Rheinboldt <10>).

Soit le point $x_{k-1} \in \mathbb{R}^n$ à l'itération k .

On tire la valeur x_k^1 de la première variable de la première équation dans laquelle les autres variables sont fixées aux valeurs $x_{k-1}^2, \dots, x_{k-1}^n$.

Puis on calcule x_k^2 dans la deuxième équation en fonction de $x_k^1, x_{k-1}^3, \dots, x_{k-1}^n$.

Et ainsi de suite.

Cette méthode est intéressante pour la résolution de modèles macro-économiques dans le cas où tous les k_i valent 1.

Vu la grande simplicité et le coût peu élevé d'une itération (les tests d'arrêt constituant les seuls calculs autres que les évaluations de fonction), cette méthode se révèle efficace pour le cas de MARIBEL¹, contrairement aux résultats des tests effectués par Gabay et al sur d'autres modèles.

Malheureusement, l'application devient difficile lors d'une exogénéisation (car les k_i ne valent plus nécessairement 1).

§ 0 LA MÉTHODE DES MOINDRES CARRÉS

Soit $F(x)$ l'application de \mathbb{R}^n dans \mathbb{R} définie par

$$F(x) = \sum_{i=1}^n f_i^2(x)$$

Il est clair que $F(x)$ est une fonction non-négative qui est nulle si et seulement si $f(x) = 0$.

Il s'en suit que la résolution $f(x) = 0$ ou la minimisation de $F(x)$ sont des problèmes équivalents.

¹ Pour le cas de MARIBEL, la méthode de Gauss-Seidel a été appliquée au système initial (sans mise sous forme quasi-triangulaire).

§ 9 ALGORITHME PROPOSÉ PAR POWELL

Powell propose un mélange entre :

- une méthode de Quasi-Newton
- la méthode de la plus grande pente appliquée à $F(x) = \sum_{i=1}^n f_i^2(x)$

Le jacobien du système est calculé par différence finie à la première itération et est ensuite ajusté par les formules de Broyden (71).

Pour la révision du jacobien, on veillera à ce que les directions d'ajustement restent "suffisamment" linéairement indépendants.

Powell tire profit de la convergence superlinéaire de la méthode quasi-Newton en combinant cette dernière avec la méthode de la plus grande pente et en introduisant un contrôle sur le pas pour assurer une convergence plus globale.

Pour ce faire, Powell introduit une variable DD, ajustée à chaque itération, qui décide du choix du pas à effectuer :

- Si la norme du pas de Newton est plus petit que DD, on effectue un pas de Newton.
- Sinon, on effectue un pas de norme DD dans une direction combinée entre le pas de Newton et celui de la plus grande pente.

L'idée de la révision de DD est semblable à celle utilisée dans notre algorithme proposé au chapitre précédent qui s'est d'ailleurs inspiré de l'algorithme de Powell.

Comme à la première itération, le jacobien est approximé par différence finie, il faut évidemment préciser la valeur du pas intervenant dans les formules (18) et (19).

Ce pas, appelé DSTEP, a également d'autres fonctions :

- Il sert comme borne inférieure à DD.
- Comme expliqué plus haut, nous faisons un "extra-step" pour la révision du jacobien dans le cas où les pas effectués ne sont pas "suffisamment" linéairement indépendants.
Cet "extra-step" se fait dans la direction "manquante" et est de norme DSTEP.
- Comme le pas de Newton et la direction de la plus grande pente sont des directions de descente de f , on estime que la valeur de la fonction doit décroître quand on fait un pas "suffisamment petit" (c'est-à-dire de norme plus petite que DSTEP).
- Lorsque le pas effectué est de norme plus petite que DSTEP, on considère qu'il n'est pas valable pour ajuster le jacobien.
Dans ce cas, on utilisera encore un "extra-step" (dans la direction du pas effectué et de norme DSTEP) pour la mise à jour du jacobien.

Dans notre version, nous avons restreint le rôle de DSTEP en introduisant une variable supplémentaire DSTEPJ.

Le calcul du jacobien ainsi que les "extra-step" s'effectuent avec un pas de norme DSTEPJ. Contrairement à DSTEP, DSTEPJ est variable et prend des valeurs entre DSTEPM et DSTEP suivant la valeur du quotient $F/PASMUL$, où PASMUL et DSTEPM sont des constantes données par l'utilisateur.

Ainsi DSTEPJ prend des valeurs relativement grandes tant qu'on est loin de la solution (F grand). Il s'en suit que dans ce cas l'approximation du jacobien donne une information plutôt globale que locale de l'allure de la fonction.

Cette astuce a pu améliorer la convergence dans le cas de MARIBEL.

Nous avons effectué de nombreux tests sur MARIBEL, mais nous nous contentons ici de fournir les résultats : les valeurs 10^{-1} , 10^{-6} et 50 semblent être les meilleures valeurs respectivement pour DSTEPM, DSTEP et PASMUL.

Une autre modification significative est l'introduction d'un "scaling" automatique (calcul automatique des facteurs d'échelle).

Sans cette mesure, l'algorithme utilisait parfois plus de 1000 itérations avant de converger (s'il convergait !).

Nous avons d'abord ramené les fonctions d'erreur sur les variables de bouclage $E_i(yb)$ à des fonctions d'erreur relative.

E_i qui était défini par

$$E_i = g_{m+i}(y) - y_{IWATEQ(m+i)}$$

devient

$$E_i = \frac{g_{m+i}(y) - y_{IWATEQ(m+i)}}{SCALE(i)}$$

où $SCALE(i)$ prend la valeur initiale de $y_{IWATEQ(m+i)}$.

Cette mesure diminuait d'un facteur 10 le nombre d'itérations nécessaires par résolution.

Comparé au Gauss-Seidel convergeant en 20 ou 30 itérations (qui était, rappelons-le, appliqué au système non réduit), le nombre d'itérations utilisées par l'algorithme de Powell était toujours très élevé.

Pour y remédier, nous avons introduit également des facteurs d'échelle sur les variables yb .

Dans l'algorithme de résolution nous travaillons avec des variables x_i définies comme suit :

$$x_i = \frac{yb_i}{SCALEX(i)}$$

où $SCALEX(i)$ prend la valeur initiale de yb_i .

L'approximation initiale est donc un point de composantes égales à ± 1 .

Ainsi, l'algorithme converge en moyenne en 10-15 itérations.

Remarque importante :

Nous tenons à signaler que malgré les mesures citées plus haut, le jacobien du système (des E_i) reste fortement déséquilibré (les valeurs des éléments varient entre 10^{-5} et 10^5).

Nous avons essayé d'autres méthodes de "scaling" pour équilibrer le jacobien, malheureusement sans succès pour la diminution du nombre d'itérations.

Il serait particulièrement intéressant d'étudier d'autres méthodes de scaling pour améliorer la convergence.

Le déséquilibre dans le jacobien peut éventuellement s'expliquer par le fait que parmi les variables de bouclage figurent aussi bien des variables économiquement importantes que des variables "bidons".

Pour cette raison, entre autres, nous proposons une recherche des variables de bouclage tenant compte de leurs poids économique (voir chapitre IV, paragraphe 3d).

Dans nos programmes, l'algorithme de Powell avec les modifications expliquées plus haut, est effectué par la sous-routine SOL.

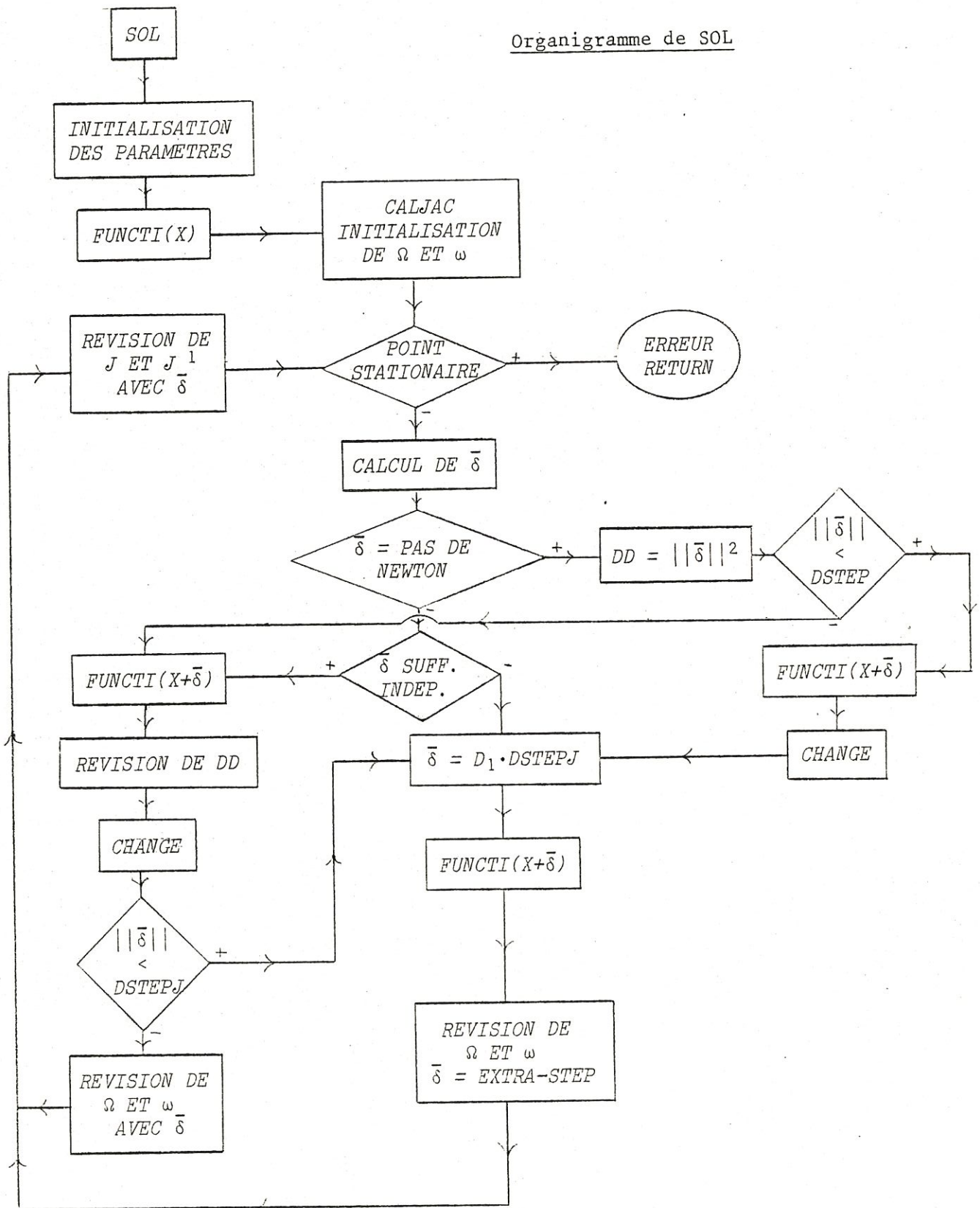
Celle-ci fait appel à plusieurs autres sous-routines :

- l'évaluation de la fonction se fait à l'aide de la sous-routine CALFUN
- l'approximation initiale du jacobien est calculée par CALJAC

Pour mieux structurer le programme, nous avons rassemblé des instructions liées dans d'autres sous-routines :

- PRINTO s'occupe de l'impression
- FUNCTI • fait appel à CALFUN pour évaluer les fonctions $(f_i), i=1, \dots, n$
 - calcule la valeur de F (FSQ)
 - effectue le test d'arrêt
 - teste si le nombre maximum d'appels de fonction permis est atteint (si oui, erreur)
- CHANGE • teste si la nouvelle approximation est meilleure que l'ancienne ; dans ce cas fait l'échange
 - effectue d'autres tests d'erreur

Organigramme de SOL



La figure 9 reproduit l'organigramme de la sous-routine SOL.

Commentaires :

- Ω et ω (WD et NWD) sont respectivement une matrice et un vecteur contenant l'information sur la dépendance linéaire des pas effectués servant lors de l'ajustement de l'approximation du jacobien.

La révision de Ω et ω est différente suivant le cas qu'on utilise un pas "normal" ou un "extra-step".

- J et J^{-1} (WJ et AJINV) sont des matrices contenant respectivement l'approximation du jacobien et de son inverse.

- $\bar{\delta}$ est le pas effectué.

- Un des critères d'arrêt est la détection d'un point stationnaire. Nous appelons point stationnaire un point à partir duquel le pas de Newton calculé est de norme plus grande que DMAX (DMAX étant une approximation généreuse de la distance du point initial à la solution).

Ce critère est vérifié entre autres dans le cas où le jacobien devient singulier.

- Dans le cas d'un "extra-step", D_1 est la direction "manquante" ; elle est calculée à partir de Ω .

Chapitre VIII

UTILISATION DES PROGRAMMES

Tous les programmes ont été faits sur le DEC 2060 des Facultés de Namur.

Nous travaillons sur le directory <MAT.PUBLIC> (MP:).

Pour exécuter nos programmes, il faut donner les instructions suivantes :

MP:BPTEMP

EXEC MP:BPMARI.FOR,MB01CD,FMO2AD,LIBFOR/SEA,MP:BPMLMO

DEL MP:BPAID*.*

où MP:BPTEMP est un programme auxiliaire pour pouvoir déterminer le temps CPU nécessaire pour la compilation du programme.

Nous utilisons des routines des librairies LIBFOR et HARWELL.

Le programme lit les données techniques dans le fichier MP:BPMLDT.DAT.

Celui-ci a la structure suivante :

ISD ISF NENDO NDON NINS NEQUA MAXLAG IEPS MAXFUN IPSER IRES DSTEP DSTEPM PASMUL
IRAVEL DRAVEL
IFILE DFILE
ILPT DLPT
DERR
TITRE
VECDON
DONNEE

Le format de lecture est :

(11I,3D,/,3(I,A10,/),A10,/,8A10,/,A10,/,A10)

Les variables ont les significations suivantes :

ISD : première année de simulation (4 chiffres)

ISF : dernière année de simulation (4 chiffres)

NENDO : nombre de variables endogènes

NDON : nombre de variables exogènes de donnée

NINS : nombre de variables exogènes de contrôle

NEQUA : nombre d'équations se trouvant dans la sous-routine MP:BPMLMO.FOR
(peut être différent de NENDO)

MAXLAG : paramètre indiquant le retard maximal dans les équations

IEPS : 10^{-IEPS} est la précision exigée pour la résolution

MAXFUN : nombre maximum d'appels de fonction permis par résolution
(sous-routine SOL)

IPSER : variable indiquant si le(s) fichier(s) de sortie doi(ven)t
contenir les données
IPSER vaut 0 ou 1

IRES : indique si le point initial de la résolution à la période t prend
les valeurs des résultats de la résolution à la période t-1, ou
les valeurs de la période t de la simulation de référence
IRES vaut respectivement 0 ou 1

DSTEP

DSTEPM sont des variables expliquées au chapitre précédent

PASMUL

IRAVEL : variable qui prend les valeurs 0 ou 1

- IRAVEL = 0 entraîne qu'il n'est pas nécessaire de faire la
renumérotation des équations. Les informations doivent alors
se trouver dans un fichier dont le nom est donné par la variable
DRAVEL (format A10).

Ce fichier doit contenir (sans format) les valeurs de L, M, S,
PERINI et SUCCES (tous entiers).

PERINI est un vecteur à 500 composantes et SUCCES est une
matrice (500,500)

- IRAVEL = 1 entraîne que l'algorithme de renumérotation est appliqué.

Les résultats seront stockés dans le fichier DRAVEL (format A10) si DRAVEL ≠ ' '.

Dans ce cas, le fichier MP:BPMLIN.DAT doit contenir les valeurs de PERINI (vecteur de permutation initiale) et la matrice d'incidence E (sans format).

PERINI et E sont entiers : PERINI(500), E(500,17).

La matrice d'incidence est stockée en bits.

IFILE : vaut 1 s'il faut créer un fichier de sortie des résultats pouvant servir comme données pour une autre simulation (0 sinon)

DFILE : nom de ce fichier (format A10)

ILPT : est l'indicateur de l'impression

0 → pas d'impression

1 → impression sur LPT

2 → impression sur LPT et dans un fichier

3 → impression dans un fichier

DLPT : nom de ce fichier (format A10) si nécessaire

DERR : nom du fichier (format A10) d'erreur

Ici s'inscrivent des informations sur les résolutions à chaque période et les erreurs éventuelles.

TITRE : vecteur double précision contenant le titre de la simulation (format 8A10)

VECDON : nom du fichier (format A10) contenant les valeurs des vecteurs IWATEQ et IDVAR (sans format).

Les rôles de IWATEQ et IDVAR, vecteurs entiers de dimension respectivement 500 et 600, sont expliquées au chapitre III.

DONNEE : nom du fichier (format A10) contenant les valeurs des variables :
1^{ière} ligne : N IDD IDF (format 3I)

où N est le nombre de variables

IDD est la première année des séries de données

IDF est la dernière année des séries de données

ligne 2 à ligne (N+1) : respectivement le nom de la variable,
son numéro, son explication, ses valeurs
pour les années IDD à IDF
(format (A6,I5,6A10,A8,40G15.6))

CONCLUSION

La réalisation de ce travail fut pour nous l'expérience la plus fructueuse de nos études. En effet, pour la première fois, nous avons réellement pu mettre en oeuvre nos connaissances et aptitudes acquises au cours de ces dernières années pour résoudre un problème se posant concrètement aux experts du Bureau du Plan. Plus d'une fois nous avons ressenti la difficulté de communication entre les utilisateurs posant le problème et les spécialistes censés le résoudre.

Habituellement confrontés à des problèmes mathématiquement bien posés, nous pensions aussi découvrir au Bureau du Plan un modèle mathématique fixe et clairement défini. Contrairement à notre attente le problème posé s'est révélé infiniment complexe, nous obligeant dès lors à nous écarter de notre but initial à savoir l'optimisation du modèle MARIBEL. Déjà pour le premier problème particulier d'optimisation, Mr. Bogaert, l'expert du Plan, nous demandait d'exogénéiser une variable antérieurement endogène. Pour des économistes, une exogénéisation est quotidienne et facilement manipulable, mais mathématiquement exogénéiser une variable revient à la considérer comme constante dans le système d'équations. En d'autres termes, pour chaque problème particulier la résolution du système doit se faire par rapport à d'autres inconnues ; ceci bouleverse évidemment toute la structure mathématique. Suite à ces difficultés, nous avons dû consacrer une grande partie de nos recherches à la mise au point d'un logiciel informatique permettant de générer un modèle mathématiquement traitable pour chaque problème économique précis.

Nous nous devons dès lors d'insister sur le fait que nous sommes conscients des lacunes et des limites de notre travail, étant donné l'ampleur de la recherche dans laquelle nous nous sommes lancés.

De nombreuses possibilités restent ouvertes pour des recherches ultérieures.

A notre avis, il serait particulièrement intéressant

- de tester et de comparer d'autres méthodes de résolution sur MARIBEL : méthode des sécantes ou méthode de Brown adaptée au problème particulier d'un modèle macro-économique
- d'introduire des poids économiques sur les variables lors de la détermination des variables de bouclage (voir chapitre IV, paragraphe 3d)
- de développer et de tester d'autres méthodes de "scaling" automatique pour la résolution du système
- de mettre au point des algorithmes d'optimisation pour MARIBEL et notamment de comparer l'efficacité de méthodes d'optimisation avec ou sans utilisation du gradient.

L'avenir répondra probablement à ces questions et nous serions particulièrement heureux de pouvoir apporter notre modeste contribution à la résolution de ces problèmes qui ont déjà suscité notre intérêt le plus vif.

BIBLIOGRAPHIE

- <1> GABAY D. ; NEPOMIASTCHY P. ; RACHDI M. ; RAVELLI A.,
"Méthodes mathématiques de la modélisation macroéconomique",
Institut de recherche d'informatique et d'automatique,
Rocquencourt 1979
- <2> GABAY D. ; NEPOMIASTCHY P. ; RACHDI M. ; RAVELLI A.,
"Etude, résolution et optimisation de modèles macroéconomiques",
Rapport Laboria, N°312, juin 1978
- <3> NEPOMIASTCHY P. ; RAVELLI A. ; RECHENMANN F.,
"An automatic method to get an econometric model in a quasi-triangular
form", Rapport Laboria, N°313, juin 1978
- <4> NEPOMIASTCHY P., "Moduleco : Software for Macroeconomic Modeling",
Institut National de Recherche en Informatique et en Automatique,
Rocquencourt, 1980
- <5> NEPOMIASTCHY P. ; RAVELLI A., "Adapted methods for solving and
optimizing quasi-triangular econometric models"
- <6> Bibliothèque HARWELL, A.E.R.E. - Report Didcot, Oxfordshire (GB)
- <7> POWELL M.J.D., "A hybrid method for nonlinear equations" tiré de <9>
- <8> POWELL M.J.D., "A fortran subroutine for solving systems of nonlinear
algebraic equations", tiré de <9>
- <9> RABINOWITZ PH., "Numerical Methods for Nonlinear Algebraic Equations",
Gordon and Breach Science Publishers, 1970
- <10> ORTAGA J.M. ; RHEINBOLDT W.C., "Iterative solution for nonlinear
equations in several variables", Academic Press, New York, 1970
- <11> CALLIER F., "Théorie des graphes", Cours de 2^{ième} licence mathématique
aux FUNDP - NAMUR

- <12> CALLIER F., "Contrôle optimal", cours de 1^{ière} licence mathématique
aux FUNDP - NAMUR
- <13> CURTIS A.R. ; POWELL M.J.D. ; REID J.K., "On the estimation of sparse
Jacobien matrices", J. Inst. Math. Appl. 13, 117-119, 1974
- <14> BAZARAA ; SHETTY, "Nonlinear Programming : Theory and Algorithms",
Wiley, New York, 1979
- <15> NOEL M.-C., "Etude, resolution et optimisation de modèles macro-econo-
miques. Application au modele belge SANDY", Memoire de fin d'études,
FUNDP - NAMUR Faculté des Sciences, 1981
- <16> BUREAU DU PLAN, Direction générale, "Le modele Maribel - Présentation
synthétique", fev. 1980 ; (80) ThD/2269/gg/2666
- <17> BUREAU DU PLAN, Direction générale, "Dossier Maribel", juin 1979,
(79) R&F-2090-pb-2381
- <18> BUREAU DE PLAN, Direction générale, "Présentation du modèle Maribel
- Son utilisation par le Bureau du Plan", jan. 1980, (80) Tdb-2246-pb-2635
- <19> INTRILIGATOR M.D., "Econometric Models, Techniques and Applications",
North-Holland Publishing Company - Amsterdam, Oxford, 1978
- <20> BROWN K.M., "Computer Oriented Algorithms for Solving Systems of
Simultaneous Nonlinear Algebraic Equations", tiré de <21>
- <21> BYRNE G.D. ; HALL CH.A., "Numerical Solution of Systems of Nonlinear
Algebraic Equations", Academic Press, 1973